



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Lauri Jaakkonen

Automated System for IEC 60598-1 Endurance Test

Metropolia University of Applied Sciences

Bachelor of Engineering

Electronics

Bachelor's Thesis

04.05.2019

Author Title	Lauri Jaakkonen Automated System for IEC 60598-1 Endurance Test
Number of pages Date	28 pages + 2 appendices 04.05.2019
Degree	Bachelor of Engineering
Degree programme	Electrical Engineering
Professional major	Electronics and healthcare technology
Instructor	Janne Mäntykoski, Senior Lecturer Aki Niskanen
<p>The purpose of this project was to create an automated test system for a luminaire endurance test, described in an IEC 60598-1 standard. The test system includes a laptop that utilizes local network, test software made with LabVIEW, and a variable AC power supply controlled via USB bus. The project was done for a client company that performs safety testing on luminaires (among other types of products) as an accredited third party. The commissioner's name is confidential.</p> <p>The project itself included the use of LabVIEW FULL graphical programming environment, learning about the IEC 60598-1 standard's clause 12.3 and interpreting it to create logical functionality for the software. Previously, the system included a lot of manual work, which was minimized with the use of laptop and LabVIEW automation, controlling a Kikusui PCR-M AC power supply.</p> <p>Besides testing the software on actual hardware after each modification, the software was regularly presented to a team of test engineers throughout its development, which may be called external validation of the software. The LabVIEW-generated software was complete well before the deadline, and there may be future updates such as updating the measurement gear.</p> <p>The result of the project was a working executable test software generated with LabVIEW, that was adopted for testing laboratory use by the commissioner. The biggest challenge for the project was learning to program with LabVIEW while also unknown errors caused by hardware led to some extra work. Using LabVIEW is just one way for creating a test software that can be used for the endurance test.</p>	
Keywords	IEC 60598-1, luminaire endurance test, LabVIEW automation

Tekijä Otsikko	Lauri Jaakkonen Automated System for IEC 60598-1 Endurance Test
Sivumäärä Päivämäärä	28 sivua + 2 liitettä 04.05.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Sähkötekniikka
Pääaine	Elektroniikan ja terveydenhuollon tekniikka
Ohjaaja	Janne Mäntykoski, Lehtori Aki Niskanen
Tiivistelmä	
<p>Tämän projektin tavoite oli luoda automaattinen testisysteemi valaisimen käyttökestävyyskokeelle, joka on kuvailtu IEC 60598-1-standardissa. Testisysteemissä on tietokone joka hyödyntää näyttökuvan jakamisohjelmaa ja LAN-verkkoa, LabVIEW:lla tehty testisofta sekä säädettävä AC virtalähde, jota ohjataan USB-väylän kautta. Projekti toteutettiin työn tilaajalle, joka tekee turvallisuustestauksia valaisimille sekä muille elektronisille tuotteille valtuutettuna kolmantena osapuolena. Tilaaja pysyy tässä työssä nimettömänä.</p> <p>Itse projektiin kuului LabVIEW FULL-versiolla ohjelmointia, IEC 60598-1-standardin pykälän 12.3 opiskelua ja sen pohjalta testisoftan toiminnallisuuden suunnittelua. Korvattava testisysteemi käyttökestävyyskokeessa vaati paljon manuaalista työntekoa, joka on minimoitu tietokoneella ja LabVIEW-automaatiolla, sekä Kikusui PCR-M-virtalähteellä.</p> <p>Testiohjelman kehitysvaiheessa toiminta oli testattavissa laitteistolla jokaisen muutoksen yhteydessä, ja lisäksi ohjelmaa esiteltiin säännöllisesti tiimille testausinsinöörejä, jotka ovat ohjelman pääasiallisia käyttäjiä. Tätä voi kutsua myös testiohjelman ulkopuoliseksi validoinniksi. Ohjelma valmistui ennen vaadittua ajankohtaa, ja tulevaisuudessa sitä on helppo päivittää, jos esimerkiksi laitteisto päivittyy.</p> <p>Projektin lopputulos oli toimiva LabVIEW:lla tehty executable-versio ohjelmasta, joka otettiin käyttöön työn tilaajan testauslaboratoriossa. Suurin haaste oli LabVIEW-ohjelmoinnin osuus. Myös muiden ohjelmoijien on helppo lukea ja päivittää ohjelmaa. Laitteistosta johtuvat tunte mattomat vikatilanteet, joista ei saatu vikakoodia ulos, teettivät jonkin verran lisätöitä projektin aikana. LabVIEW-ohjelmointi on vain yksi tavoista luoda hyväksyttävä testiohjelma valaisinstandardissa kuvatulle käyttökestävyyskokeelle.</p>	
Avainsanat	IEC 60598-1, käyttökestävyyskoe, LabVIEW-automaatio

Table of contents

Abstract

Table of contents

Abbreviations

1	Introduction	1
2	Theory and Components for the Project	1
2.1	Background Information	1
2.1.1	Devices	3
2.2	IECEE and Standards	6
2.2.1	IEC Luminaire Standards	7
2.2.2	Clause 12.3: Endurance Test	7
2.3	LabVIEW Basics	8
2.3.1	User Interface	9
2.3.2	Execution Structures	10
2.3.3	Data Structures	11
2.3.4	SubVIs	12
3	Programming of Endurance Test software with LabVIEW	13
3.1	Summary of the Endurance Test software	13
3.2	Electrical Output Modes	14
3.3	Electrical Output's On and Off Periods	16
3.4	Program Structure	17
3.5	Data Logging	19
3.6	ActiveX Automation with LabVIEW	21
3.7	Pause and Continue	24
3.8	E-mail Notifications	25
3.9	Remote Monitoring of Endurance Test	26
3.10	Future Updates	27
4	Conclusion	28

Appendices

Appendix 1. Block Diagram

Appendix 2. Front Panel

Abbreviations

AC	Alternating Current
CPU	Central Processing Unit
DC	Direct Current
EN	European Standard drafted and maintained by CEN
GPIO	General Purpose Interface Bus
I	Current
IEC	International Electrotechnical Commission.
IEC 60598-1	An IEC Standard that defines general requirements and tests for luminaires.
IECEE	IEC System of Conformity Assessment Schemes for Electrotechnical Equipment and Components
LED	Light Emitting Diode
NI	National Instruments
P	Power
PID control	Proportional-Integrational-Derivative control.
RAM	Random Access Memory
RGB	Red-Green-Blue
RS-232	Recommended Standard 232

SMTP	Simple Mail Transfer Protocol
T	Time
TDMS	Technical Data Management System, file type designed by NI
U	Voltage
USB	Universal Serial Bus
VI	Virtual Instrument; a functional building block for LabVIEW code or a virtual instrument program created with LabVIEW.
VISA	Virtual Instrument Software Architecture

1 Introduction

The test software created during this project was made for a commissioner that performs testing and certifying on luminaire products (among other types of products) according to multiple certification schemes.

Luminaire department of the commissioner performs a variety of tests described in generally accepted standards, such as EN and IEC publications. Typically, luminaire manufacturers pay for these tests in order to receive a certificate that proves the product's conformity to a set of standards. Standard series 60598 includes many publications, and these set requirements and tests for different types of luminaires. The first part of the series, IEC 60598-1, sets general requirements and tests for luminaires, and it includes electrical, and mechanical tests and visual inspections. The test studied for this project is a luminaire endurance test, which is defined by clause 12.3 of IEC 60598-1.

The test software created during this project is referred to as Endurance Test software. It was created using LabVIEW graphical programming environment after studying and interpreting clause 12.3 to create logical functionality for the software. The test software controls a Kikusui PCR-500M variable AC power supply via a USB bus and uses it for measuring electrical values while controlling its output. The software also fetches information from a website on local network and uses Microsoft Outlook for creating automatic test notifications. For remote monitoring the endurance test, an open-source screen sharing program was used. The test software is designed for test engineer use.

2 Theory and Components for the Project

2.1 Background Information

This project started by acquiring a copy of IEC 60598-1 edition 8.1 to study. Luminaire standards are under constant change due to technical advancements, mostly with LED

technologies. Clause 12.3 of IEC 60598-1 covers tests for newer (LED) and older types of luminaires, such as filament lamps, and all possibilities are considered in this project.

The commissioner of the project provided a LabVIEW FULL version for the project. LabVIEW FULL version includes more features than the basic version; some of these are remote program controlling, PID control, and signal processing and control. With LabVIEW license also comes access to technical support, software updates and online training. National Instruments' forums and example finder are also helpful; most LabVIEW virtual instruments have examples that can be easily studied or modified for one's own use. Support and examples for using hardware with LabVIEW can also be found online for thousands of different devices.

Before the introduction of the Endurance Test software created during this project, the test is done manually with a custom-made wall cabinet manufactured in year 1992 that has the following functionalities: variable 3-channel electrical output, electric timer for turning voltage output on and off, and ambient temperature measurement. The ambient temperature is also measured by another sensor, which uploads measurement data to an Environment System - web page, from where the temperature and humidity can be monitored on local network. The units under test are behind a glass panel in a small room that has its temperature controlled. Meanwhile, an Agilent 2534 Digital Power Meter measures and displays electrical values on its front panel.

The wall cabinet with integrated electrical outlets and electrical timer can be replaced by a laptop and a programmable power source. The new system with remote monitoring and notifications eliminates the need for walking to measurement site, requires less space and provides a better user experience with a custom software made specifically for test engineers. Data logging is also more consistent.

The goal was to create an automated test setup that logs data, is remotely monitored and interactive to user in case the subject is failing the endurance test. The purpose was to save time, allowing a test engineer to allocate work time and effort for other tasks while an endurance test is running for seven or ten days. Measurement data is available in one computer file and is collected more frequently, compared to the old way of collecting measurement data manually.

After consulting the team of test engineers, it was agreed that the company's Outlook can be used as a platform for the notification system. Outlook is an e-mail service offered by Microsoft. LabVIEW provides functions for automatic e-mail messaging using an SMTP server. Optionally, test engineers may use a separate open-source application on computer for remotely monitoring the test program. Later, a USB controlled RGB-LED status indicator will be added to the test setup, so the test software may indicate test status both on - and offline.

2.1.1 Devices

2.1.1.1 Test Computer

A Lenovo ThinkPad with an Intel i5-processor and eight gigabytes of RAM was used during the project. LabVIEW-generated programs work different when it comes to other programming languages; for example, a LabVIEW program automatically allocates and frees memory, whereas for most text-based programming languages, memory must be allocated at the start of the program. LabVIEW-generated programs can be run on simple computer systems, if infinite loops and uncontrollably growing arrays (or other memory issues) can be avoided, and, if a lot of processing power is not needed. No memory issues were faced during the creating and testing of the Endurance Test software.

The executable endurance test software issues commands to a remote controllable power supply each second, does simple data type conversions and comparisons with the measured data and keeps time with system clock. It also saves measurement data to disk every 10 minutes by opening, writing to and then closing the measurement file. The heaviest task for the program is at the end of the test, where all the measurement data of a TDMS file is read at once and graphs for temperature, voltage and current are generated. The graphs are then saved in picture format on an Excel file. Typical values of CPU and RAM usage are shown in figure 1:

The screenshot shows the Windows Task Manager Performance tab. The 'Performance' tab is selected, and the 'App history' sub-tab is active. The main area displays a table of resource usage for the currently selected application, '12.3 (32 bit)'. The table has columns for Name, CPU, Memory, Disk, Network, GPU, and GPU Engine. The application is using 2.2% CPU, 52.6 MB of memory, 0 MB/s of disk, 0 Mbps of network, and 0% of GPU. The overall system usage is shown at the top: CPU 28%, Memory 38%, Disk 18%, Network 0%, and GPU 1%.

Name	CPU	Memory	Disk	Network	GPU	GPU Engine
Apps (4)						
> 12.3 (32 bit)	2,2%	52,6 MB	0 MB/s	0 Mbps	0%	

Figure 1 Windows Task Manager and resources used by the test software.

As figure 1 shows, the memory usage can be compared to that of a web browser. Also, with the laptop's display in power saving mode the test computer should remain rather power efficient.

2.1.1.2 Kikusui PCR-M

Kikusui PCR-M is a computer-controllable, variable AC power supply. A Kikusui PCR-500M was used, which respectively has output power of 500 watts. It comes with remote control option via a USB or RS-232 bus, and newer models can even be controlled via ethernet cable. Kikusui PCR-M power supply is controlled using NI-VISA commands; NI-VISA is a standard that provides a programming interface between hardware and software, and it functions with the USB, ethernet, serial, GPIB, and few other interfaces. [3]. NI-VISA or Kikusui - provided VISA drivers are required of the computer for programmatically controlling PCR-M voltage source.

The PCR-M accepts commands in string form and all the necessary commands are provided in manuals provided by Kikusui. All that is required from LabVIEW-made program for communication with the PCR-M is to open a reference to a VISA resource, and then use VISA functions such as VISA Write or Read. Updating to newer Kikusui PCR model or switching bus is relatively easy, as only some of the string commands (if any) would have to be replaced.

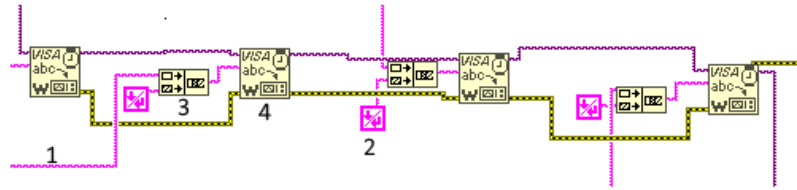


Figure 2 Example of using VISA Write VI

As figure 2 shows, string commands for Kikusui PCR-M are passed along pink wires (1) to a concatenate string function (3), where an end of line constant (2) is added to the end of the string command. The concatenated string is then delivered to a VISA Write object. The whole string is read by the PCR-M as a single command. Reference to VISA resource is passed along the purple wires. Data flow is handled with wiring, and therefore VISA Write functions (4) execute in order from left to right. This is a snippet from the test software.

For the endurance test, AC or DC mode is selected by the user on the user interface of the test software. The AC/DC selector chooses between two cases in a case structure that has the appropriate string commands for these modes, as shown below in figure 3.

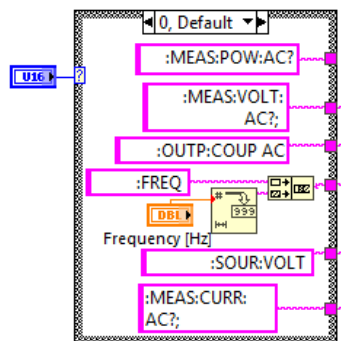


Figure 3 Initial string commands for PCR-M

The case visible here is for the AC output mode. Numerical value for frequency is converted into string form and concatenated with `:FREQ` string. A measurement made on the PCR-M takes about 330 milliseconds to complete, so instead of using `:MEAS:POW:AC?` string command and making the PCR-M busier, the power is calculated by the computer in real time with the test software. Measuring voltage and current

with hardware provides all of the necessary information as Power equals Voltage multiplied by Current. String command `:MEAS:CURR:AC?` gives the RMS-value of current. These string commands are then passed around the main Block Diagram via wires to their respective VISA Write or Read – functions.

2.1.1.3 Temperature Measurement

The client company has two options for measuring the temperature during endurance test:

- temperature-controlled cabinet equipped with a thermocouple
- National Instruments-USB TC01, a USB device reading a thermocouple

National Instruments USB TC-01 device connects to a USB bus and can be used with most thermocouple types. LabVIEW has example programs available for using this device and integrating it to the project would be simple. The temperature-controlled test cabinet's measured temperature and humidity can be monitored on the company's Environment System web page accessible via Local Area Network. LabVIEW can fetch this data by reading the HTML code as an array of strings. This is how the current version of Endurance Test software works.

Both available temperature systems include the use of thermocouples; these are simple and cost-efficient sensors for measuring temperature. A Thermocouple typically consists of two metals that are welded together at what is called a junction point. When temperature changes, a voltage is created at the junction point, which can then be measured. Different types of thermocouples (for different types of metals used) have different effective temperature range, durability, chemical resistance, and other attributes. [4].

2.2 IECEE and Standards

IEC System for Conformity Assessment Schemes for Electrotechnical Equipment and Components (IECEE), is a certification system based on IEC standards [14]. International Electrotechnical Commission (IEC) globally publishes international standards and

manages conformity assessment systems. IEC focuses on electrotechnology fields, and their publications serve as a globally accepted basis for standardization. [11].

Global IECEE schemes (CB and CB-FCS), which are based on the IEC standards, help in reducing trade barriers caused by different certification criteria in different countries. Need for multiple testing is eliminated, allowing products to reach market while lessening manufacturing costs. [15].

2.2.1 IEC Luminaire Standards

IEC 60598-1 is the first part of 60598 series of standards, setting general requirements and tests for luminaires. The standard includes requirements and tests related to luminaire safety and performance. Tests described in the standard are type tests; *“test or series of tests made on a type test sample, for the purpose of checking compliance of the design of a given product with the requirements of the relevant standard”* [2,24]. A luminaire is generally tested as the product is delivered, and installed similar to the normal use of the luminaire. Generally, testing is made on one sample, and it shall be safe to operate after tests to comply to the standard’s requirements. [2, 15].

IECEE lighting standards (60598 series) include parts for more specific types of luminaires beside the first part, such as 60598-2-1 (normal luminaires for fixed installation) or 60598-2-5 (floodlights), e.g. There are over 20 second parts for the 60598 series. Typically, the first parts of standards are used in conjunction with a relevant, more specific part of the series while testing products, and for example, IEC 60598-1 gives permission to use any applicable part as required for the testing.

2.2.2 Clause 12.3: Endurance Test

Endurance test is on for seven or ten days, depending on luminaire type. The test subject is mounted in a draught-proof thermal enclosure for the test. Changes in temperature are avoided, so the enclosure’s temperature must be controlled. The ambient temperature for testing changes depending on the luminaire, but unless specified by manufacturer, the testing temperature is held within two Celsius degrees of 35 degrees Celsius. During the test, a test subject goes through multiple on and off-periods in higher than

normal operating voltage, up to 168 hours or 240 hours. A single cycle consists of 21 hours on and three hours off periods, where the on-phase begins a cycle. For LED luminaires, a test period is ten cycles and for filament/other type of luminaire seven, where the last cycle is an abnormal condition phase that typically requires hardware modifications on the test subject.

Test voltages vary for different types of luminaires, and generally have a specific multiplier for operating voltage depending on the luminaire type; for LED luminaires, 1.1 times the operating voltage is used, whereas filament luminaires are operated by keeping the power constant and fine-adjusting voltage. For measurement accuracy, the luminaire's supply voltage must remain within one percent of the test voltage. [2,95]

Other requirements, such as abnormal conditions where faults are simulated, are specified in clause 12.3, and in support of enabling manual work to be done mid-test to the subjects, manual and automatic pausing together with continue button were implemented. Typical LED luminaires are not tested in abnormal condition.

2.3 LabVIEW Basics

LabVIEW is a graphical programming environment and is used for creating Virtual Instrument applications, VIs. These VIs may be exported as executable programs for any computer. Programming with LabVIEW is called graphical programming, because the code itself is mostly objects in loops and data flows via wires. Support for importing bits of code from text-based programming languages, such as C, Python or Java is also found.

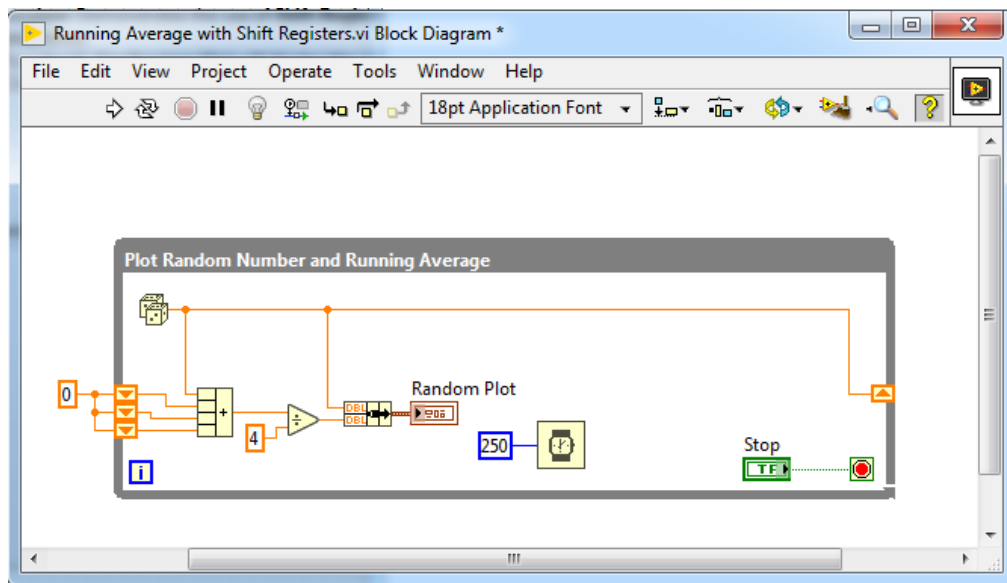


Figure 4 Logic diagram of LabVIEW program

The LabVIEW application of figure 4 calculates the running average value of four most recent random numbers. The code is in a loop that iterates every 250 milliseconds, and three previous values are stored in Shift Registers (orange arrows). This is an example program called Running Average with Shift Registers.vi, provided by the LabVIEW example finder. Objects on logic diagram are created by simply right-clicking and selecting an object from Functions – palette.

LabVIEW is popular because creating simple applications is relatively fast. Reading the graphical code is intuitive and most programs' code can be fitted on one computer display. On top of that, LabVIEW automatically allocates and refreshes memory, making it one less thing for the programmer to worry about. LabVIEW supports thousands of devices, providing drivers and fully functional example programs. Using devices with no LabVIEW drivers at hand requires a means of connecting them to a machine running LabVIEW and having a datasheet or a manual that provides control commands. In the case of this project, VISA commands were used via a USB bus.

2.3.1 User Interface

LabVIEW has a Front Panel and Block Diagram window for each virtual instrument application. Block Diagram is where the graphical programming takes place while the Front

Panel window is for creating a custom interface for one's application and includes, for example, controls, buttons and indicators. Figures 4 (above) and 5 provide demonstration of the interface.

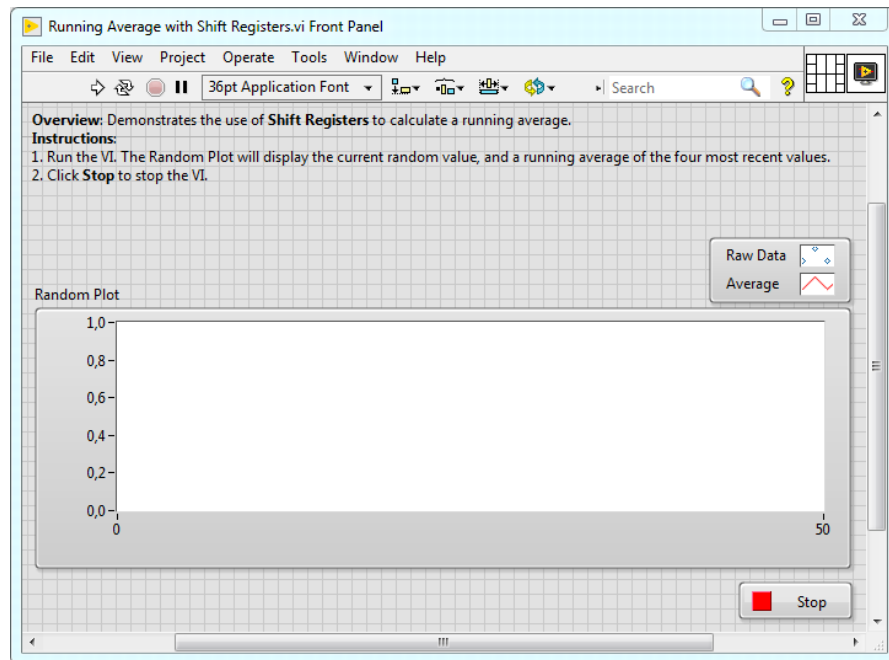


Figure 5 Front panel of Running Average with Shift Registers.vi

Objects on the Front Panel are freely customizable by the programmer, and both Block Diagram and Front Panel allow text boxes for commenting on the code or instructing to use the program.

2.3.2 Execution Structures

Execution structures have graphical code in them, and the type of structure varies how or when the code inside is run. Most common examples would be executing code repeatedly, or when meeting a certain condition such as a read value from a measurement device. Multiple execution structures may be assigned order of execution by simply utilizing dataflow of wires.

The execution structures used in making the test software are While loops together with Event and Case structures. An Event structure waits for an event to happen (such as a button press or a mouse click), then executes the code inside it; for example, the press of a button causes the start time of endurance test program to be printed on the front panel. Case structure may have multiple states that execute under different conditions. By default, a Case structure has a True and False case selectable with Boolean value, and more than two cases can be managed by using additional objects such as creating a number from multiple Booleans and passing the numerical value to a case selector.

A While loop is similar to a do loop or repeat-until loop in text-based programming languages [10]. While loop executes all code in it repeatedly until it is stopped by sending a Boolean True to its Conditional Terminal.

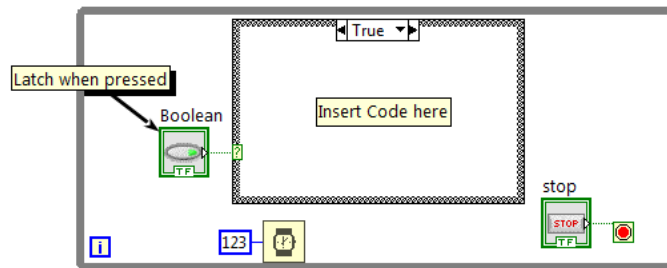


Figure 6 Case Structure inside a While loop

A while loop that is allowed to execute continuously with no time delay between iterations will use a lot of processing power of the computer. Therefore, a Time Delay VI is put inside the while loop. This makes the while loop wait a set amount of time between its iterations. The program presented on figure 6 executes the While loop every 123 milliseconds. When the Boolean button (that is set to latch when pressed) gives a Boolean True, the code inside the Case structure executes once. Otherwise the program would execute any False condition of the case structure, which may also be left empty.

2.3.3 Data Structures

LabVIEW has String, Numeric, Boolean and Dynamic data types. Wires and objects are always associated to a data type. These are portrayed by different colors representing

different data types, easing the readability of logic diagrams. Data types may be converted to different types using conversion objects, while directly connecting objects of different data types result in a broken wire, indicating the program cannot execute.

Related data may be grouped together using arrays and clusters. Arrays combine multiple points of one type of data type into one data structure, while clusters combine the data points of many data types into a data structure. An array is made of elements and dimensions. For example, multiple subsequent data points (elements) that have a voltage reading and its respective timestamp are used for creating a one-dimensional array. [12].

2.3.4 SubVIs

A LabVIEW application or just part of the code may be converted into a SubVI that requires less space on the block diagram (figure 7). After creating a SubVI it can be used on another VI's or SubVI's block diagram. There is no limit to the hierarchical number of these layers. [13]. This modularity helps managing the code and is one of the good aspects of LabVIEW. A SubVI works the same way as a VI, but it is called from within the main VI. It resembles a subroutine for text-based programming languages.

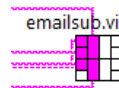


Figure 7 SubVI of graphical code that sends e-mail notifications

A SubVI also has a block diagram and user interface, which may be quickly accessed from within the main VI by right-clicking and choosing Open Front Panel. A SubVI also has all the necessary inputs and outputs that can be found in its original code (figure 8). These are automatically assigned when creating a SubVI.

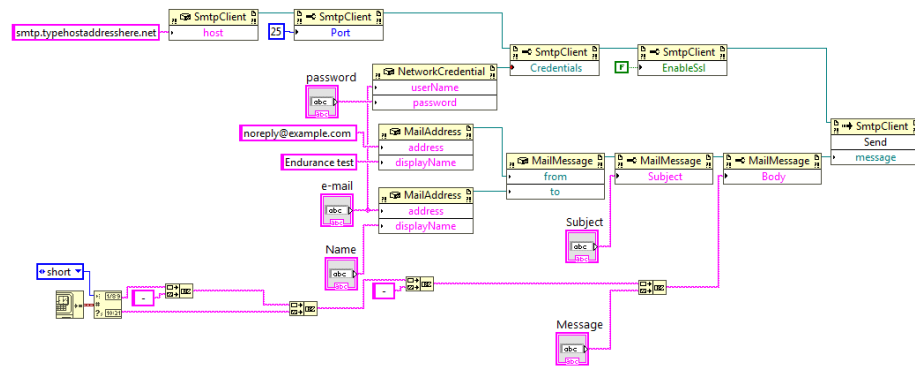


Figure 8 Block Diagram of the SubVI

Here the difference between readability and size of the graphical code can be compared. The SubVI object on figure 7 is the size of a string control object on figure 8 (purple squares).

3 Programming of Endurance Test software with LabVIEW

3.1 Summary of the Endurance Test software

This chapter presents multiple key aspects of the finished Endurance Test software by reviewing some parts of the graphical code. The test software was created with LabVIEW FULL version and an executable application was built of the virtual instrument program. This application may be run on any computer that has IVI-COM drivers by Kikusui, and NI-VISA (or Kikusui VISA) Runtime Engine and LabVIEW Runtime Engine from National Instruments.

The team of test engineers was consulted multiple times throughout the development phase. This helped in improving the software in the best way for the user by implementing requested features and doing other adjustments to the project. Some of these features are an automated e-mail system regarding test status, a remote monitoring view of the laptop screen that is accessible with an URL and a status indicating RGB-LED light, which will be implemented later.

Caption of the Front Panel of the endurance test-program can be found in Attachments. Using the software requires the user to:

- Set electrical output values and choose AC or DC mode
- Define number of cycles and times for on and off - periods
- Set data logging sample rate and choose whether an Excel file is saved
- Choose a file path for measurement files
- Type any measurement information required for raw data file
- Enable e-mail notifications (optional).

The front panel of Endurance Test.exe is also used for monitoring the measurement, showing a real time graph of electrical power, time, cycle number, humidity and temperature. Advanced Settings allow enabling of constant power mode.

3.2 Electrical Output Modes

Luminaires for the test are operated with either constant current, voltage or power. The PCR-M has a constant voltage output, but software utilizing its output measurement functionality and a Proportional-Integral-Derivative (PID) control can use the PCR-M as a constant power source. Standard 60598-1 states that, for example, filament or other types of luminaire that require constant power operation, the operating voltage should be 1.05 times the voltage at which rated wattage of the luminaire is obtained.

PID control is a commonly used control algorithm, and widely accepted in industrial use [1]. A PID control reads a measured variable and adjusts the output of another variable which, again, affects the measured input. The input gains (figure 9) for PID control in LabVIEW have effect on how much or how fast the PID control adjusts the output. The best values were chosen after manual testing, and in this case derivative time was not used.

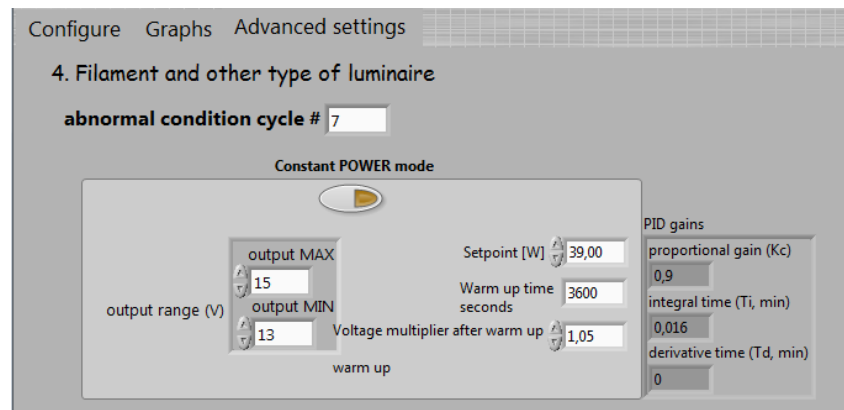


Figure 9 Constant power mode under advanced options

Constant power mode is located under Advanced settings. Because the standard 60598-1 defines that the initial on-period includes the warm up-period of a filament luminaire, the user has the option of defining a warm up time for luminaire. After the warm up time, the output voltage gains a voltage multiplier by which the luminaire is to be operated through the test, and PID control turns off.

PID control was also tested for constant current mode but was then removed after initial testing. The IEC standard 60598-1 states that a luminaire without control gear, for constant current operation, shall be operated with 1.1 times the rated current. An LED light source without a control gear was used to test a simple PID control. Forward current was specified as 450 mA, with a maximum value of 600 mA. IECCE Operational Document 5014 defines accuracy limits for laboratory instruments, and in this case the limit is +/- 1.5%. Here are the results showing Kikusui voltage output and the corresponding current to the LEDs.

Table 1. Output current from Kikusui PCR-M to an LED light source without control gear.

Output voltage (volts)	Measured Current (amperes)
14.5	0.17
15.0	0.399
15.2	0.41
15.4	0.41
15.6	0.41

15.9	0.42
16.0	0.67

The LED light source was meant to be operated at 0.66 amperes. The table shows that trying to reach a 1.5% accuracy with Kikusui on low voltages is difficult due to the 0.1 volts resolution of Kikusui PCR-M's output. Not reaching the desired current using the PID control in this case will cause the output current to alternate between too high and too low values, and so it was concluded with the engineers that variable current source shall be used manually, if a similar test situation must ever be implemented.

3.3 Electrical Output's On and Off Periods

Programmatically the output's on and off-switching and cycle counting were implemented in a While loop that works independently next to a While loop that handles all data measurement and logging. These loops might resemble a producer/consumer structure which is a commonly used application structure. The elapsed time value is constantly compared to user-defined values for on and off-periods respectively, and a Boolean True or False to a Selector object dictates when the case changes. When switching between on – and off-cases, a Boolean True is sent to the Reset-input of Elapsed Time.vi, and the timer starts from zero for the next on or off-period. These phases alternate for the duration of the Endurance Test, until finally the cycle iteration count has reached its limit.

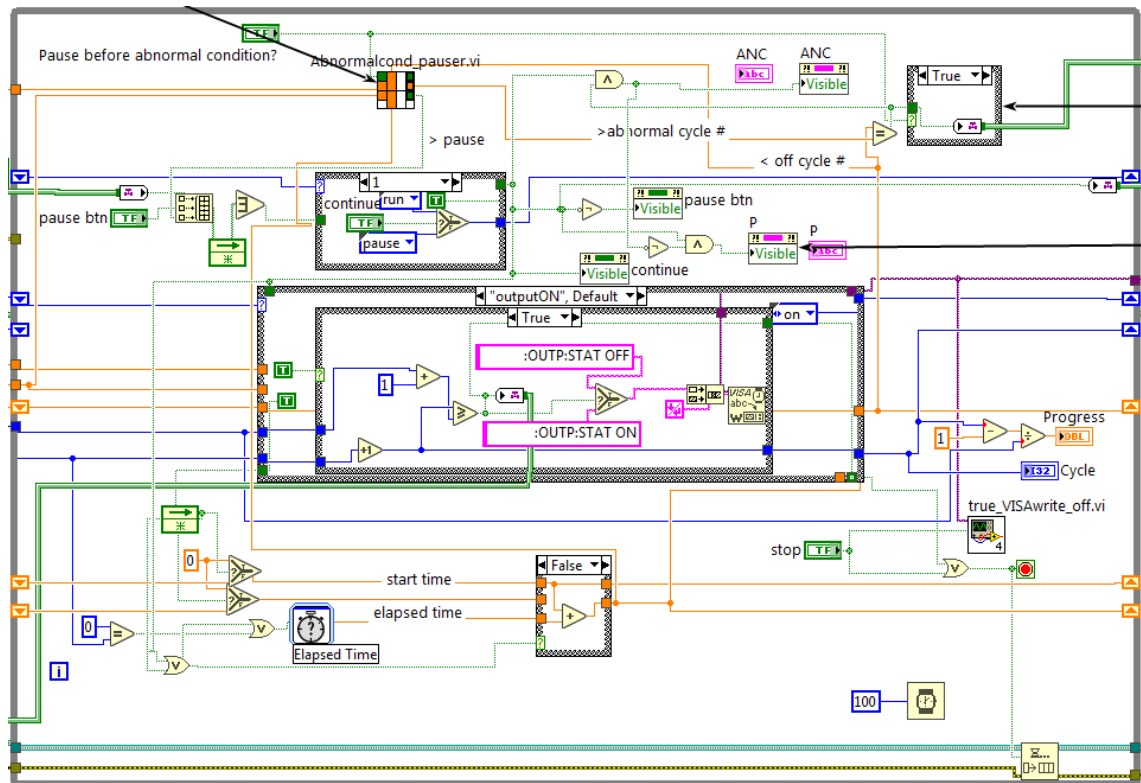


Figure 10 While loop that keeps time, handles pausing/continuing and switching of electrical output

The Block Diagram in figure 10 presents some of the main functions for controlling the Kikusui PCR-M. Some sub-vis, elapsed time, and logic behind pausing and continuing can be seen.

When the set number of test cycles are completed, a Boolean True value from the loop is delivered via a queue to stop the measuring loop. Asynchronous loops work simultaneously, if there is no data dependency between them and the occasional data flow between is handled via queues or tag channels.

3.4 Program Structure

All the code is in one while loop, so that the test program runs when opened. This allows the user interface buttons to be interactive (hide/show buttons or refresh VISA resource, e.g.). When the user presses Start, the actual endurance test executes once; the program runs through each step of a state machine, and then stops.

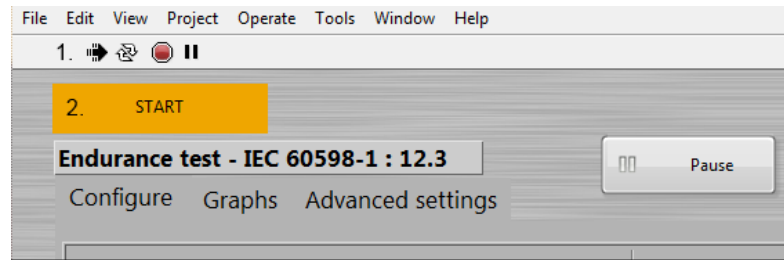


Figure 11 Endurance test software running when launched

In figure 11 the black arrow (1.) indicates a running program. On the executable version, the user is not required to use this row of buttons, and the row may be hidden. Customized start-button (2.) starts the execution of the actual test after the user has set up parameters for the measurement.

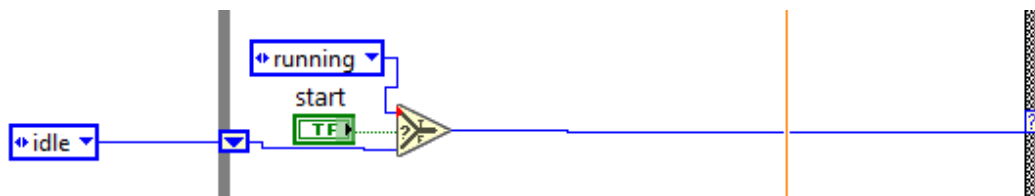


Figure 12 State machine inside a while loop

In figure 12 Enumerator Constant is used to set initial value of the case structure. The while loop will stay in idle mode, where the Front panel buttons are active but the actual code that performs the Endurance Test will not run until the Start button is pressed. When the selector reads a Boolean True, running-state is selected, and the program will go through the rest of the state machine's states in order.

The second step (running) includes something that resembles a producer-consumer loop; two asynchronous while loops where one of the loops takes measurements from device and local network (temperature and humidity) and the other while loop only turns on and off the voltage source, until set amount of cycles (seven or ten days for actual endurance tests) have passed. Pausing and continuing is implemented for the second while loop, and Boolean True values for different pause conditions are passed between the while loops via tag channels. These Boolean values dictate the pause message shown for the user and any e-mail notification that is sent using the company's SMTP server.

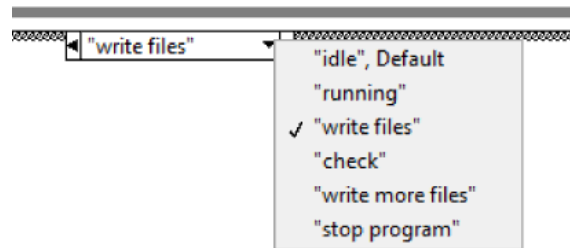


Figure 13 Steps of the main program's state machine

Different steps are shown in figure 13. In the third step of the state machine, the TDMS file containing all measurement data is opened, and graphs for temperature, voltage and current values are saved. The graphs get starting time from an event structure that triggers when start button is pressed to run the endurance test.

In the fourth step, the program checks whether files by the name of graph files have been created. This will result in a Boolean true to case selector, which will allow the program to proceed to the fifth step of the state machine for writing Excel files. Otherwise the program will continuously continue to check whether the image files exist yet. This should prevent any error of ActiveX automation trying to open a file that does not exist or is still being created by the computer.

The fifth step of the program is for ActiveX automation for Excel file. It programmatically utilizes a TDMS-Excel addin to be able to open the TDMS file of the endurance test. Temperature, voltage and current graphs are added on the first worksheet, and finally the excel file is saved in the folder of the TDMS file. The excel file then includes all the measurement information, and a glance at the graphs on the first worksheet will present all the necessary information. All measurement data points can be found as text columns on a "Measurement Data"-worksheet.

3.5 Data Logging

Company refers to data collected from measurements as raw data. Company instructions (that are based on standard IEC 17025) state that raw data must include:

- Sample number
- Test operator's name
- Project number
- Date
- Which test is performed
- Environmental factors
- Inventory numbers of measurement equipment.

The test software lets the user choose a sampling rate for data collection. Available choices are every second, every minute or once every 10 minutes. The test engineers were consulted, and a sampling rate of 10 minutes was agreed on being the default value for longer tests. On diagram level, the measurement loop has a one second iteration time, so a quotient and remainder function is used to send a true Boolean every 600th iteration, which is 10 minutes in this case. Every time a Boolean True is read, the measured values of that moment are written to a TDMS file.

The program writes measurement data on a TDMS file, which is a LabVIEW format that stands for Technical Data Management Solution. TDMS is a format with hierarchical organization structure consisting of file, group and channel levels. All the data is stored in one file. Low level TDMS functions (figure 14) were used instead of express VIs, providing more control over the software.

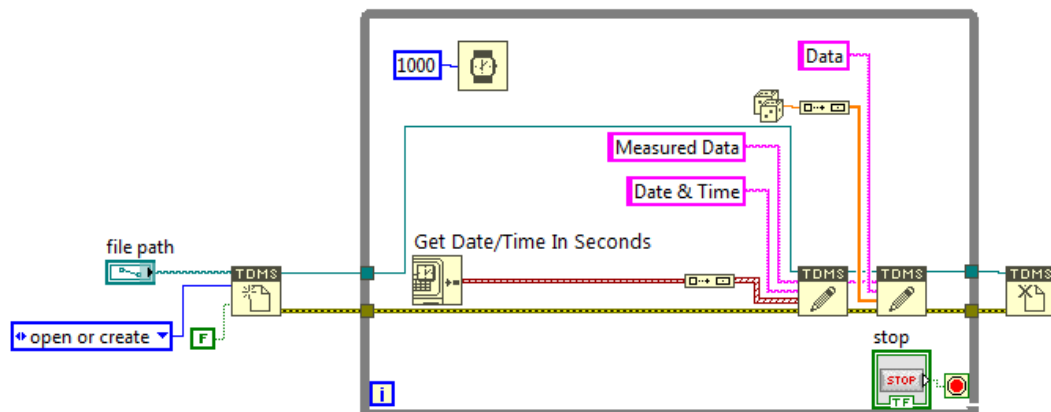


Figure 14 Example of writing measurement data with TDMS functions.

In figure 14 a TDMS file in a specific file location is first opened or created (if it does not exist initially), and after that the while loop operates. Inside the loop are two TDMS write objects. Data is written under a 'Measured Data' group name, which consists of time and data channels. The dice represent random data generated which is delivered to TDMS Write as an array. The TDMS file closes after the loop is stopped by user interaction.

The test software is designed for a long test, and precaution against an error or a power outage is taken. The TDMS file is opened, written on, and closed every 10 minutes so that the measurement file will always have the latest data on disk. Only after the test is complete, the software opens the TDMS file in Excel using ActiveX automation tools and a TDMS to Excel – plugin. Using ActiveX automation methods, three graph pictures are saved on the TDMS file on a new worksheet and the file is saved in excel format.

3.6 ActiveX Automation with LabVIEW

Measurement data for all the tests done in the company is available in excel format. LabVIEW supports ActiveX automation, which is a set of Microsoft Technologies that enables the linking of individual programs – in this case, LabVIEW and Excel. For this purpose, National Instruments offers a Report Generation Toolkit for a price of 574.3€ (1/2019). LabVIEW may also be used as an ActiveX automation client, which is the free approach.

LabVIEW provides help for using ActiveX automation (figure 15) such as examples and topics on how it works.

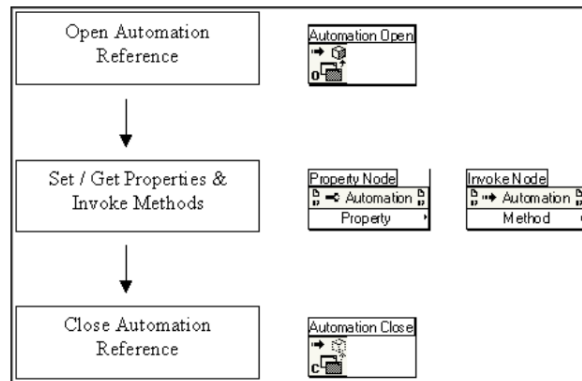


Figure 15 ActiveX automation on LabVIEW

Figure 15 shows how ActiveX automation generally works [9]. An automation reference to excel is opened, after which single commands such as creating a sheet, or adding data, is performed by using property nodes and invoke nodes. It is always recommended to close automation references when they are no longer needed.

This project's ActiveX part was done after following multiple examples, troubleshooting errors caused by mismatching Excel versions and folder permissions. The ActiveX part of this project currently opens a TDMS file using the required Excel addin, creates a new sheet on Excel for graph images and adds the graphs to the worksheet. The TDMS file path is automatically taken from the measurement data save path, and the .xls-file is created to the same folder. Figure 16 demonstrates importing TDMS to Excel – addin using ActiveX automation.

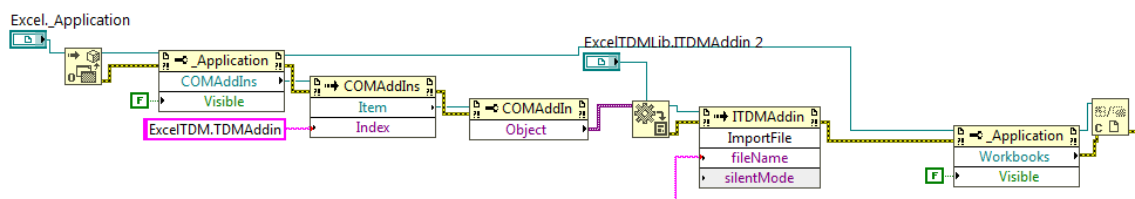
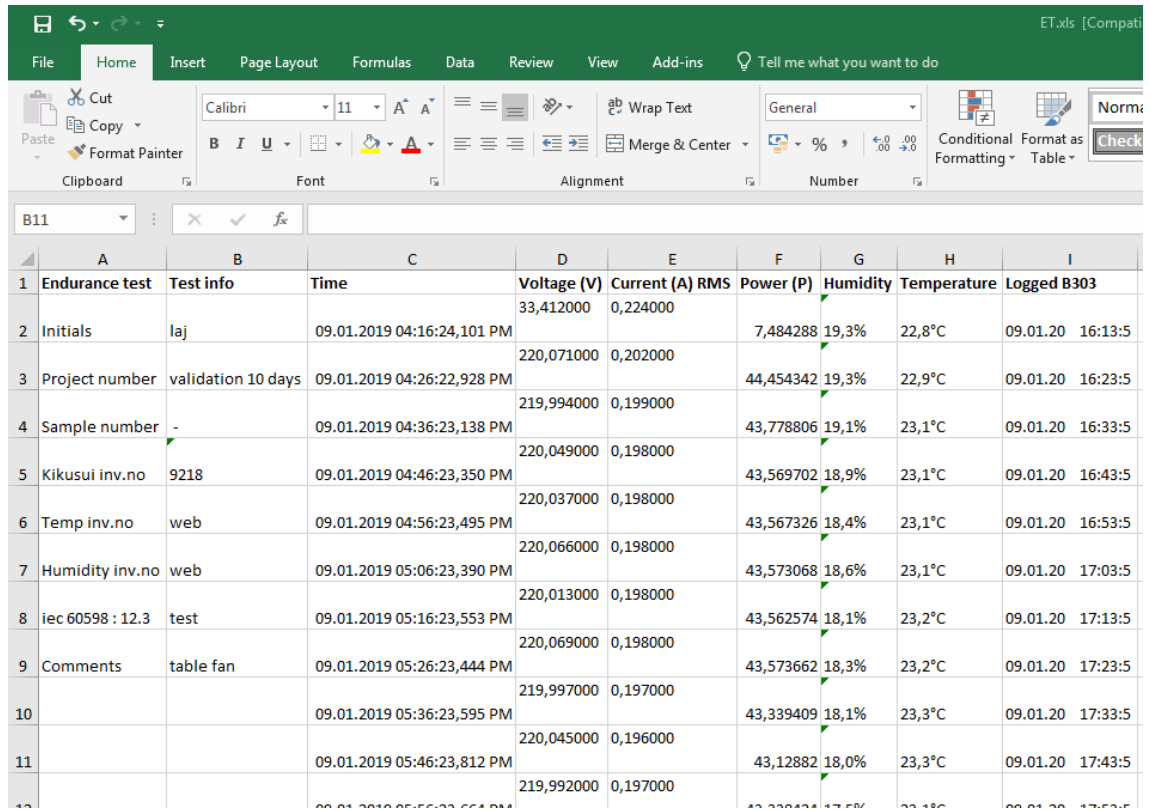


Figure 16 Part of code that utilizes the ActiveX automation.

After opening Excel automation reference, the TDMS Addin is changed from variant to data LabVIEW can utilize and finally a TDMS file is opened at a file location that is brought via the pink wire to 'fileName'-input of ITDMAddin-node.

ActiveX's Property and Invoke nodes would cause unknown errors that did not generate any specific error code. This was found on multiple property and invoke nodes and the

problem could be fixed by removing and again placing identical objects back on the diagram. Excel should be set to not update automatically to prevent some of these problems. The ActiveX automation code works and provides results as shown in figure 17 below.



	A	B	C	D	E	F	G	H	I
1	Endurance test	Test info	Time	Voltage (V)	Current (A) RMS	Power (P)	Humidity	Temperature	Logged B303
2	Initials	laj	09.01.2019 04:16:24,101 PM	33,412000	0,224000	7,484288	19,3%	22,8°C	09.01.20 16:13:5
3	Project number	validation 10 days	09.01.2019 04:26:22,928 PM	220,071000	0,202000	44,454342	19,3%	22,9°C	09.01.20 16:23:5
4	Sample number	-	09.01.2019 04:36:23,138 PM	219,994000	0,199000	43,778806	19,1%	23,1°C	09.01.20 16:33:5
5	Kikusui inv.no	9218	09.01.2019 04:46:23,350 PM	220,049000	0,198000	43,569702	18,9%	23,1°C	09.01.20 16:43:5
6	Temp inv.no	web	09.01.2019 04:56:23,495 PM	220,037000	0,198000	43,567326	18,4%	23,1°C	09.01.20 16:53:5
7	Humidity inv.no	web	09.01.2019 05:06:23,390 PM	220,066000	0,198000	43,573068	18,6%	23,1°C	09.01.20 17:03:5
8	iec 60598 : 12.3	test	09.01.2019 05:16:23,553 PM	220,013000	0,198000	43,562574	18,1%	23,2°C	09.01.20 17:13:5
9	Comments	table fan	09.01.2019 05:26:23,444 PM	220,069000	0,198000	43,573662	18,3%	23,2°C	09.01.20 17:23:5
10			09.01.2019 05:36:23,595 PM	219,997000	0,197000	43,339409	18,1%	23,3°C	09.01.20 17:33:5
11			09.01.2019 05:46:23,812 PM	220,045000	0,196000	43,12882	18,0%	23,3°C	09.01.20 17:43:5
12			09.01.2019 05:56:23,664 PM	219,992000	0,197000	43,338424	17,5%	23,3°C	09.01.20 17:53:5

Figure 17 Measurement results in xlsx – format

Graphs and measurement data are included in the file. Because of trying to avoid excessive use of ActiveX automation, 'test (root)'-worksheet was not removed. These are the results of a week-long test, where everything worked as intended.

The implementation of data logging using ActiveX automation functions is time consuming and prone to errors. Due to unknown errors it was best to do as little with ActiveX automation as possible. It would be easier to only log data to text files, TDMS files and images. LabVIEW has some express VIs that save measurement data with minimal amount of coding, such as 'Write to Measurement File VI'. However, express VI usage does not give the programmer as much control over the program as lower level VIs do.

3.7 Pause and Continue

It is considered a good practice to create a pause and continue functions for a program. Endurance Test software is paused either manually or by the program when certain conditions are met. Pausing is implemented in the timer while loop that functions as an on-off-phaser for Kikusui PCR-M. Pausing also serves as a safety mechanism, since it is good to turn off the electrical output of a voltage source to prevent a fire hazard, for example. When the program is paused for any reason, the output voltage is turned off, the timer loop pauses, and the measurement loop writes "PAUSE" on measurement data columns.

Time Elapsed express VI was used with shift register and a case structure. The Time Elapsed.VI does not support pausing but can be reset, so the elapsed time is stored in a Shift Register. Pausing causes the Shift Register value to stay unchanged (by keeping the last value in the shift register through consequent iterations) and pressing continue resets the Elapsed Time.VI with a Boolean True. The time value from Shift Register is then added to the zero-value that is read from Elapsed Time.VI. To the user, this seems like a pause instead of resetting and adding numbers.

Pausing was made interactive to the user by creating an e-mail notification and a message window on the user interface. The message windows are actually string controls that have instructional text for the user in different cases; the text is set as default value and the actual input field of the controls are disabled from user. The string control windows were customized to have yellow coloring with red borders, making it more noticeable to the user. A Display Message VI could have also been used, but it would halt the program while waiting for user action - using a string indicator is simpler code.

Respectively, property nodes are used for changing front panel buttons' visibility, making the program more intuitive to use. Pause and continue buttons cannot be seen simultaneously and start-button disappears from the user once the test starts.

3.8 E-mail Notifications

The information that the test software receives is either electrical values from measurement device or data from local network, and any notifications or pausing is performed based on the electrical values. Should current or voltage reach unwanted values during on or off-period, the program will automatically pause and send an e-mail.

User may enable an e-mail system with a simple button located on the program's user interface, as shown in figure 18. The program sends e-mail messages using the company's SMTP server.

2. E-mail notification

Enable e-mail notifications? OFF

- remote monitor URL
- Current over safe limit
- Program finished
- Abnormal condition cycle coming up

I min during ON (A)

From

Subject

E-mail address

Password

Remote monitor URL

Figure 18 A snippet from the user interface.

Here the user may also define a minimum current limit when the program shall send a notification. The program calculates a running average of latest and previous iterations' values for current to reduce erroneous notifications. The USB bus reads 0.003 amperes as a zero with the inductive load that the software was tested on, so 0.008 is just an example value near zero. The notification is sent when output voltage is on but no current flows; this could happen when an electrical component inside a luminaire breaks, for example.

National Instruments provides a working example for using LabVIEW to send e-mail messages using SMTP server (figure 19).

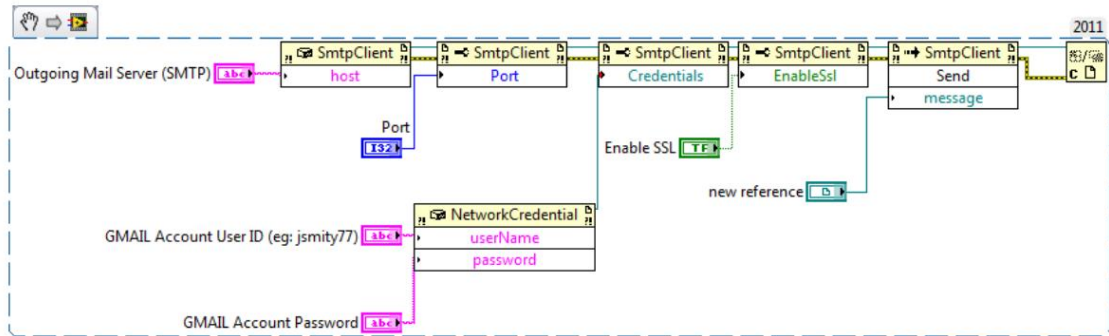


Figure 19 A LabVIEW example for sending e-mail messages [5].

For the above code to work, the computer must be connected to a network and have the required privileges to send e-mails. This was achieved by defining a port and, in cooperation with the IT department, giving the laptop permission to create and send e-mail messages using SMTP server. By using case structures, First Call? – vis and Boolean Trues for different cases, the program successfully sends customized notifications in different scenarios such as when the program is finished or automatically paused. The First Call vis are used to avoid spam messaging.

3.9 Remote Monitoring of Endurance Test

Remote monitoring is one of the functions that was initially deemed necessary. Remotely controlling a voltage source over network could be a safety risk and is forbidden by the client company rules. Therefore, options for only sharing screen were considered. There are a few alternatives such as Windows Remote Desktop and TeamViewer, which were ultimately scrapped. Windows Remote Desktop would require some modifications to make it share screen only, without giving access to control.

Ultimately, an open source application (by Eslam Hamouda) called ScreenTask was used. ScreenTask is a screen-sharing program that creates a webpage to local area network. The computer acts as a server and uploads a screenshot to the webpage in customizable intervals. The URL address can be attached onto the e-mail notifications, making the test monitoring accessible in one step, even on a mobile device.

The only downside with ScreenTask is that it must be launched manually. However, no software installation by the client is required. The rate of uploading the screenshots can be changed to reduce strain on the host computer. [6].

3.10 Future Updates

It is easy to integrate or change code on the current logic diagram of the software due to LabVIEW's modular nature. Any LabVIEW-compatible measurement device may be used to take the measurements, and a standalone measurement software for taking electrical and other measurements will be implemented in the future. This will be done using a Yokogawa WT series digital power meter, for which LabVIEW drivers are available.

Considering the programming, the use of local variables is avoided as it creates risks of losing data between asynchronous loops that have different execution speeds. [7]. This is exactly the case in this project, where the asynchronous while-loops for timing and measurement resemble a producer-consumer design pattern. Instead of using local variables, it is better to use a tag channel or a queue; both transfer data between two asynchronous loops without making them data dependent of each other. Figure 20 shows the implementation of a tag channel.

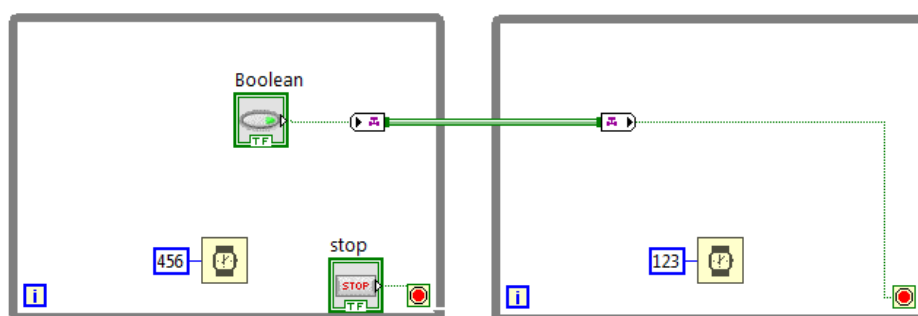


Figure 20 Tag channel

In figure 20 a Boolean pushbutton value (True or False) is written to a tag channel by a writer endpoint. It is then delivered via the tag channel to a separate while loop, where a reader endpoint gives the Boolean output of the pushbutton to the loop's Conditional Terminal. Tag channel contains a single value and shares it to any number of reader

endpoints. The channel is always refreshed to show the latest value. The two loops now have no data dependency between each other and work asynchronously. [8].

Temperature measurement system is not final, and currently the information is read from web. Temperature measurement will be taken from the website or with a USB device that reads thermocouples.

Besides automated e-mail notification system, for locally indicating the test's status there will be a USB-controlled RGB-LED status indicator created by Luxafor. It is powered by the USB bus and easily operated with LabVIEW using Python commands, which LabVIEW supports.

4 Conclusion

This project was finished with a working product that was successfully validated for the client company to use. Throughout its development, the endurance test software was being constantly validated. Each added function was tested on actual hardware, and the team of test engineers was regularly consulted to confirm the direction of development. Measured values and timing accuracy of test computer were finally validated with a 10-day test run, simultaneously confirming the readings using external, calibrated devices.

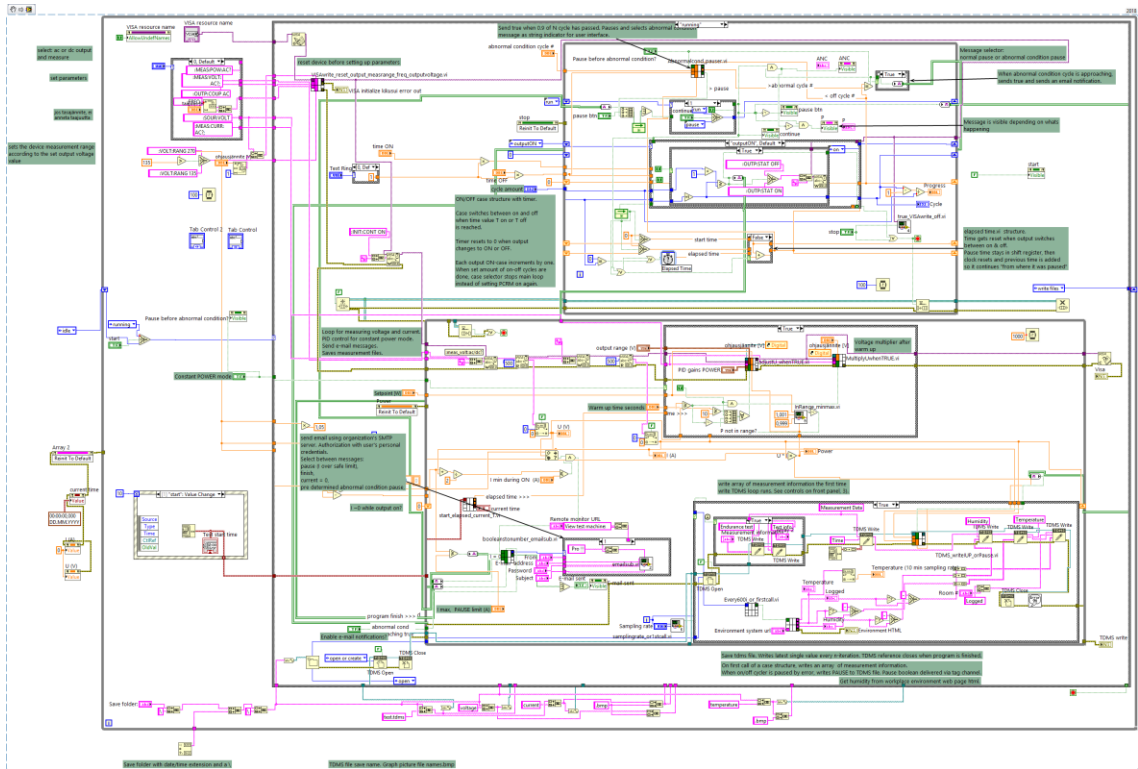
Future updates may include the option of choosing different measurement gear, i.e. a different power supply, or a separate measurement device when the electrical output is controlled by something other than Kikusui PCR-M. Next step after the project is keeping up the software, installing it on any required measurement setups and solving any problems or bugs that may surface. Parts of the code may be re-used in future LabVIEW projects.

Reference List

1. National Instruments White Paper about PID theory. [Online] URL: <http://www.ni.com/white-paper/3782/en/>
2. IEC 60598 – 1 edition 8.0. General requirements and tests for luminaires.
3. Description of NI-VISA by National Instruments. [Online] URL: <https://www.ni.com/visa/>
4. Information about thermocouples. [Online] URL: <https://www.thermocoupleinfo.com/>
5. National Instruments article about sending e-mail using SMTP server with LabVIEW. [Online] URL: <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019LNrSAM>
6. Website of open-source screen-sharing program ScreenTask. [Online] URL: <https://screentask.me/>
7. National Instruments, example about local variables. [Online] URL: <http://www.ni.com/example/28717/en/>
8. National Instruments, article about tag channels. [Online] URL: http://zone.ni.com/reference/en-XX/help/371361N-01/glang/channel_tag/
9. National Instruments White Paper about ActiveX with LabVIEW. [Online] URL: <http://www.ni.com/white-paper/2983/en/>
10. National Instruments article about LabVIEW execution structures. [Online] URL: <http://www.ni.com/getting-started/labview-basics/execution-structures>
11. About International Electrotechnical Committee. [Online] URL: <https://www.iec.ch/about/activities/?ref=menu>

12. National Instruments article about Data types and structures. [Online] URL:
<http://www.ni.com/getting-started/labview-basics/data-structures>
13. National Instruments tutorial for using sub-VIs. [Online] URL:
<http://www.ni.com/white-paper/7593/en/>
14. About IECEE. [Online] URL: <https://www.iecee.org/faq/>
15. About IECEE. [Online] URL: <https://www.iecee.org/about/what-it-is/>

Appendix 1. Block Diagram



Appendix 2. Front Panel

