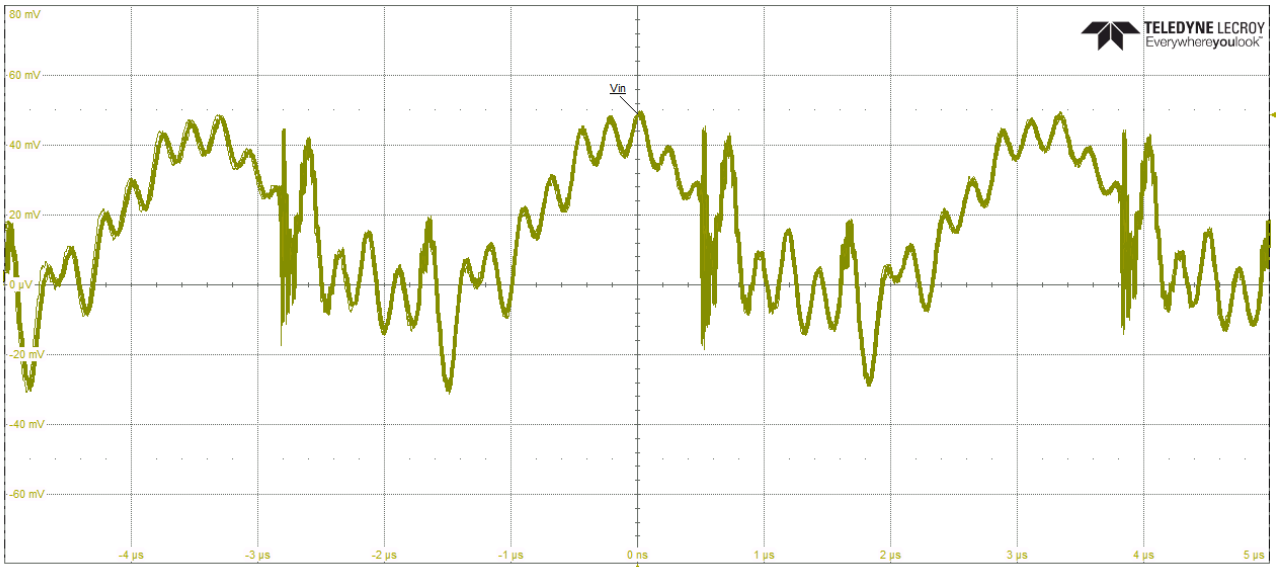
CHALMERS



Evaluate and improve automatic testing on printed circuit boards

A report on the development of automatic testing

Bachelor thesis in Mechanical Engineering

NOAH LANAI

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Bachelor thesis 2021

Evaluate and improve automatic testing on printed circuit boards

NOAH LANAI

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Evaluate and improve automatic testing on printed circuit boards
Bachelor's degree project in Mechanical Engineering
NOAH LANAI

Supervisor: Henrik Isaksson, Ericsson
Examiner: Bengt Lennartson, Department of Electrical Engineering

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96, Gothenburg
Telephone + 46 (0)31-772 1000

Cover:
Capture of voltage signal on LeCroy WaveSurfer 4034HD

Typeset in LaTeX
Gothenburg, Sweden 2021

Acknowledgements

The degree project was carried out in the Power Solution laboratories at Ericsson in Gothenburg, Sweden.

First and foremost I would like to express my gratitude to my supervisor Henrik Isaksson for the unparalleled support and guidance on the project. Secondly, I would like to thank my examiner Bengt Lennartson for the continued support and feedback on the thesis. I would like to extend my sincere thanks to my manager Mattias Saxlöv for offering me the project.

I would also like to thank Fredrik Larsson for the advice and constructive criticism. I would gratefully acknowledge the assistance of Jörgen Jansson for his ideas and interest in the project. I also had the pleasure of working with Stefan Bryne.

At last, I would like to thank everyone else who contributed to the project.

Noah Lanai, 2021

Abstract

Testing and verification of hardware at Ericsson is up to date performed by manually controlling electronic instruments. The internal software EICE, Ericsson Instrument Control Environment, can create scripts that control the instruments automatically. This project created a concept of a solution that automates six of the test parameters used for verification. The solution was built around the verification Excel file, which specifies the system requirements for the hardware. The work was done experimentally by programming and developing solutions in EICE. After the automatic solution was built, it was analysed and the new testing method evaluated. The solution was more than three times as fast as the manual method. It also lets the developer work on other important tasks while the tests are performed automatically. However, a problem was found that prevented the solution from working properly. To fix this, a software update to EICE has to be done. If the issue was to be resolved, it would certainly be possible for a general automatic testing method to function. There are still ideas and improvements that could be made to the scripts if there is time and effort. The project has contributed to making the verification Excel file more suitable for automatic testing. It has also contributed to fixing two bugs with EICE.

List of acronyms and definitions

AI	Artificial Intelligence
EICE	Ericsson Instrument Control Environment
Eload	Electronic load
N/A	Not Applicable
Osc	Oscilloscope
PCB	Printed Circuit Board
Pload	Pulse load
SAU	Support Alarm Unit

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter defines the purpose and aim of the project.

1.1 Background

Ericsson is a multinational networking and telecommunications company that offers services, software and infrastructure in information and communications technology for telecommunications operators, network equipment and mobile broadband. The Power Solution Laboratories, which are responsible for the research and development of the network hardware is located at Ericsson Lindholmen. Developers work on innovating and finding new solutions to improve the antennas and modules powering radio, baseband and transport networks. The hardware must be able to withstand different environments and situations caused by nature. This implies for instance that it can continue working during power grid failures for some time. To ensure that the hardware is reliable and robust, engineers test and verify that it meets the system requirements. These tests are performed in the laboratories by connecting instruments to the printed circuit boards and measuring how they react to different currents, voltages and temperatures. Up to date, the tests are performed by controlling the instruments manually. With the use of the internal software EICE, they could be automated. This would make testing more standardized and save developers plenty of time.

1.2 Objective

The purpose of this project is to build a general automatic solution that automates six of the test parameters used for performance and robustness verification. This solution will make testing more standardized. If everyone would perform the tests with the same method, it could decrease the risk of errors.

1.3 Delimitations

All of the test parameters used for performance and robustness verification will not be covered. Six test parameters will be enough to get a grip of the impact that will come from changing methods. The automatic solution will only work for one converter at a time. There exists other software for automating and testing hardware. EICE will be the only software used for automation in this project. When evaluating the testing method, the project will not cover investments or ecology aspects but will focus primarily on the time saved.

1.4 Related work

Developers have built scripts in EICE to automate tests. These scripts were made for automating separate test parameters. Their solutions are not general and most of the data needs to be changed manually. These scripts are written in previous versions of EICE that was not compatible with Microsoft Excel. Therefore they are outdated. These scripts can be looked at for tips and inspiration when building the new general automatic solution.

1.5 Research questions and contributions

The following questions were the topics researched in the project. The questions are brought up as conclusions in Chapter 6.

- Is the current version of EICE capable of creating a general automatic testing method that is functional?
- How much time is saved by performing the tests automatically?

The project has contributed with a concept of a general automatic testing method that is more than three times as fast as the manual testing method. An issue with the volt/div setting in EICE was found that prevents a general automatic testing method from working properly. Two bugs were found during the verification of the automatic solution. These bugs were fixed by the software developer. The thesis has also contributed to making the verification Excel file more suitable for automatic testing.

1.6 Outline

Chapter 2 is an introduction to electronic testing at Ericsson and gives an overview of the tools and instruments used for verification. Chapter 3 explains the methodologies used for building the automatic solution and for evaluating the new testing method. Chapter 4 presents how the automatic solution is built. The structures of the scripts are presented with Microsoft Visio flow

charts. Chapter 5 explains the evaluation of the new testing method. Chapter 6 ends the thesis by answering the research questions and proposing future work.

Chapter 2

Overview of verification, testing and EICE

This chapter is an introduction to electronic testing at Ericsson.

2.1 Verification process

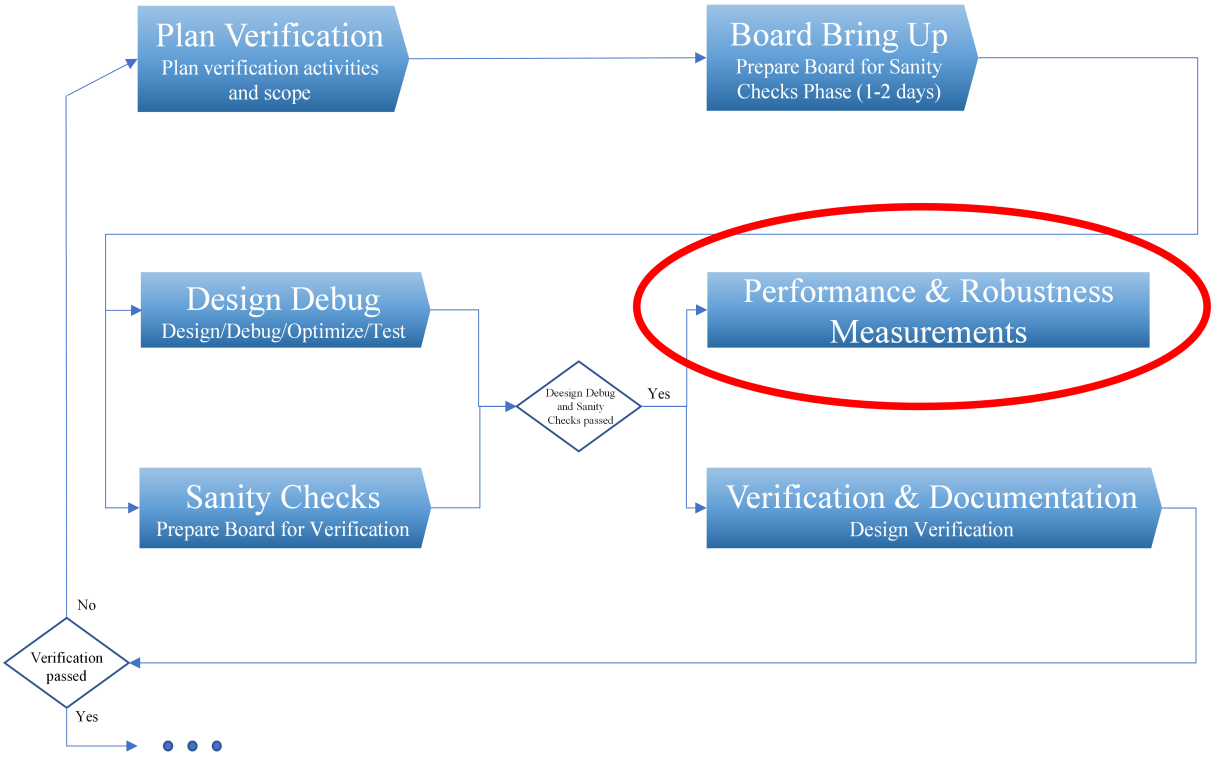


Figure 2.1: Power Solutions - Board Power Verification Process Overview

All of Ericsson's products are tested to verify that they meet the system requirements. For a product to pass the verification it has to go through all the steps shown in Figure 2.1. The last step of the verification process is the performance and robustness measurements. (Ericsson Internal, n.d.)

2.2 Testing procedure

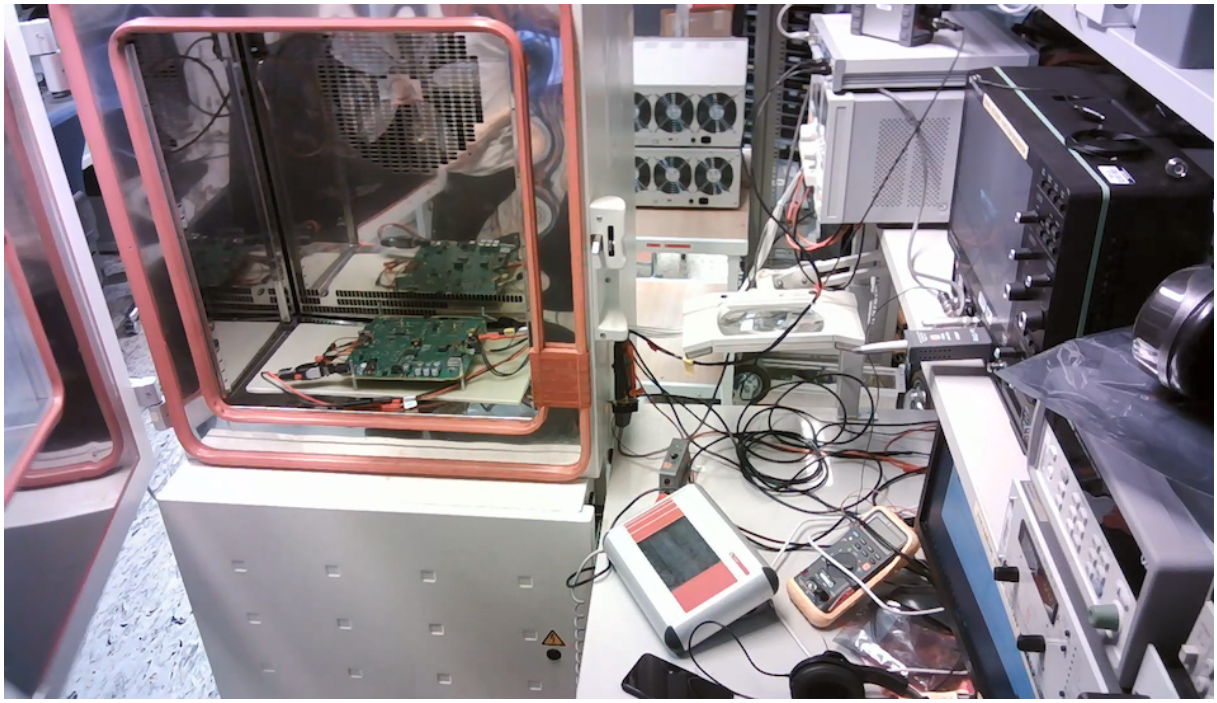
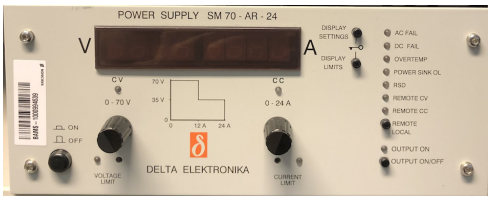


Figure 2.2: Test-rigg

The performance and robustness measurements are performed in the laboratories by connecting electronic instruments to the printed circuit boards and measuring how they react to different currents, voltages and temperatures. The instruments used for testing are a power supply, an electronic load, a pulse load, an oscilloscope, a Bode100 and a temperature chamber. The power supply is the device that supplies electric power to the printed circuit board. The electronic load is used to sink the current and absorb the power of the printed circuit board. The pulse load is used to bring a current slew rate of 2.5A/µs. The pulse load is explained in-depth in Section 2.5. The oscilloscope is used to perform different measurements of the current- and voltage signals. The Bode100 is used to measure the phase margin, gain margin and cross over frequency of the electronic circuit. The temperature chamber is used to heat or cool the test environment. Figure 2.2 shows an example of the setup. The printed circuit board is placed inside the temperature chamber and is connected to the electronic instruments. (Bryne, S. 2019)

2.3 Manual testing

(a) LeCroy 4034HD Oscilloscope



(b) Delta Elektronika SM-70-24 Power Supply

Figure 2.3: Physical control panels

Up to date, the tests are performed by manually controlling the instrument's physical panels or their corresponding digital panels in EICE (Section 2.4). Figure 2.3a shows the physical panel of an oscilloscope and Figure 2.3b shows the physical panel of a power supply. The developer screws and fidgets with the buttons on these panels to change the settings. After correctly setting up the conditions for a test parameter he/she has to manually change the oscilloscope settings to get a measurement of the specific test parameter. At last, the results have to be manually documented in Excel. This means that every person in the lab will perform the tests in their own way and at their own pace. Performing a full rail of verification manually takes about 8 hours (a workday). (Bryne, S. 2019)

2.4 EICE

EICE, Ericsson Instrument Control Environment, is a software that allows the user to control electronic instruments via the computer. It presents digital versions of the instrument's physical control panels. This gives the developer an overview of the panels. The digital panels in EICE let the developer do everything that the physical panels could with some limitations. The most useful function in EICE is the ability to write scripts. The scripts create a sequence of instructions for controlling the instruments with basic functions such as loops, if-then conditions, calculations and subroutines. EICE can therefore be used to automate the testing process. EICE can also communicate with Microsoft Excel which enables the result documentation to be automated as well. (EICE, 2020) (Ericsson internal, 2017)

2.5 Pulse Load

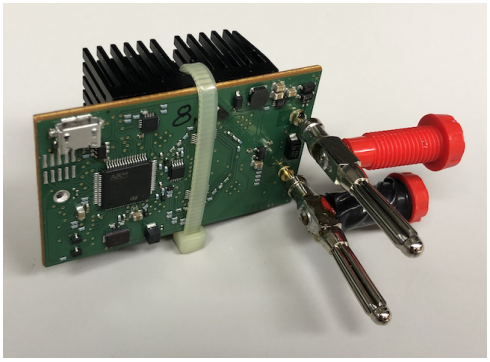


Figure 2.4: Pulse load prototype (Rosén, L. Sahar, S. 2020)

One of the test parameters requires a current slew rate of 2.5A/µs. The electronic loads used in the laboratories cannot produce a slew rate of this speed with the supply voltages required for the tests. Because of the electronic load's dimensions, it needs cables to connect to the printed circuit board. These cables affect the current slew rate. This problem is solved by using a pulse load that is connected closer to the printed circuit board. This pulse load can bring a slew rate up to 32A/µs. Figure 2.4 is a picture of a pulse load. (Rosén, L. Sahar, S. 2020)

2.6 Verification sheet

The system requirements for the performance and robustness measurements are specified in an Excel file. Each product has different system requirements. Therefore, the excel file is split up and organized with a sheet for every product. Each sheet contains a list of all the system requirements for each specific product. (Bryne, S. 2020)

Template name	Example
Description	Verification of product x
Procedure	Measure using instruments x1, x2 and x3
Pre-condition	For requirements 1234 - XX123
Status	x

	Vin	55	1%
	Vout	5	0.5%
Input Data	x	x	x
Voltage	x	x	x
	x	x	x
	x	x	x

	Nominal load	6,25	0%
Input Data	x	x	x
Current	x	x	x
	x	x	x

Test case		Condition	Pass criteria		Measurement at temperature			Result
Requirement tag	Parameter		Min	Max	-40°C	+25°C	+100°C	Pass/fail
		Min load @ Vin.min	54	56				-
		25% of Imax (1,56) @ Vin.min	54	56				-
	Test 1, Vin.min	50% of Imax (3,13) @ Vin.min	54	56				-
		75% of Imax (4,69) @ Vin.min	54	56				-
		100% of Imax (6,25) @ Vin.min	54	56				-
		Min load @ Vin.nom	54	56				-
		25% of Imax (1,56) @ Vin.nom	54	56				-
XX-YY-xxx	Test 1, Vin.nom	50% of Imax (3,13) @ Vin.nom	54	56				-
		75% of Imax (4,69) @ Vin.nom	54	56				-
		100% of Imax (6,25) @ Vin.nom	54	56				-
		Min load @ Vin.max	54	56				-
		25% of Imax (1,56) @ Vin.max	54	56				-
	Test 1, Vin.max	50% of Imax (3,13) @ Vin.max	54	56				-
		75% of Imax (4,69) @ Vin.max	54	56				-
		100% of Imax (6,25) @ Vin.max	54	56				-

Figure 2.5: Verification sheet, simplified

Figure 2.5 shows a simplification of the Excel sheet that specifies the system requirements for one of the products developed at Power Solutions. The sheet consists of three main sections. The first one gives general information about the product and the verification. The middle section specifies the input data used for performing the tests. This input data tells the developer which values and units will be used. The last section specifies information about the tests. It tells the developer what test parameters to perform and under which conditions. It shows the pass criteria for each test and a place to write the results. The last section contains the input data for the temperatures. That is because the hardware is always tested for three temperatures. (Bryne, S. 2020)

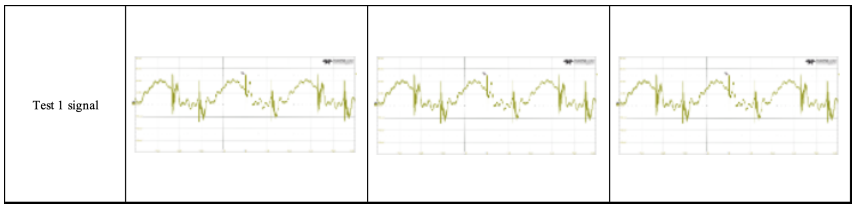


Figure 2.6: Picture output, example

Some of the test parameters require pictures of the electronic signal(s). The pictures are required for certain test conditions. The cells to put those pictures are located below the information of the tests. Figure 2.6 shows an example of how the pictures are set in the verification file.

2.6.1 Conditions

The tests are performed with a specific amount of conditions. The number of voltage levels and current levels that are used for the tests depends on the product and the test parameter (Bryne, S. 2020). Looking at Figure 2.5, the number of different tests the product has to go through can be calculated. For each temperature, the product is tested with minimum-, nominal- and maximum voltage. For each voltage level, the product is tested with five load levels. This gives the developer 45 conditions for Test 1. The developer will perform Test 1 under 45 different conditions. The results of the tests are placed in the cells below the temperature input. For some products and tests, it might only have for example 36 test conditions.

2.6.2 Pass criteria

Pass criteria		Measurement at temperature			Result
Min	Max	-40°C	+25°C	+100°C	Pass/fail
54	56	53	55	55	Fail
54	56	57	53	55	Fail
54	56	55	55	55	Pass
54	56	N/A	55	N/A	Pass
54	56	55	N/A	55	Pass

Figure 2.7: Results of tests, example

Figure 2.7 is an example of the results after a test. For the product to pass the test, the result must be within the boundary given by the min and max criteria. If the result falls within the limits, it will pass the test. If the result falls outside the limits, the product will not pass the test. Some of the tests are not prioritized or not needed for certain products to be considered reliable. These tests will display N/A, which means “Not Applicable”. It tells the developer to skip those tests. (Bryne, S. 2020)

2.7 Summary of overview

The performance and robustness measurements are performed in the laboratories by connecting electronic instruments to the printed circuit board and measuring how it reacts to different currents, voltages and temperatures. Up to date, testing is performed by manually controlling the electronic instruments. EICE is the software used for automatically controlling the instruments. In some tests, a Pulse Load needs to be used. The information of the tests is specified in the verification Excel file. The different sheets specify the system requirements for each product. The different products have different test conditions and pass criteria.

Chapter 3

Method

This chapter explains the methods used in the project.

3.1 The ideal solution

The ideal solution would be a script that completely runs through the entire verification. It would automatically perform all of the test parameters and publish the results in the right places. It would be a general solution that works on every product sheet. The user would only need to input what product is being tested and EICE would run the automation by itself. The ideal solution would be made in such a way that the script can be easily readable and understandable by the developers using it. In that way, workers can enter and change the code if needed. This solution would have functions for troubleshooting every possible fault in the measurements. That way, the automation would continue to run during instrument failure. The ideal solution would be so safe and reliable that it can be run during nighttime without monitoring.

3.2 Building a general automatic solution

The solution was built entirely around the verification Excel file. It was constructed to cover four of the product sheets. The four product sheets looked different and specified the system requirements for four different products. The idea was that the solution would work no matter which of these sheets you chose. The product sheets were the following:

- Primary requirements
- Switch requirements
- Linear requirements
- Sau requirements

There were two noticeable differences between the test conditions. These differences were taken into account when building the solution. The differences were the following:

- Primary requirements did not require to test output voltage ripple for 25% load and 75% load
- Sau requirements did only require testing with nominal voltage.

The solution covered six test parameters used for verification. The six different tests were made as separate scripts to be implementable in a script that puts them together. The script that puts the tests together communicated to them via global variables. The order of the test scripts was set in consideration to the four product sheets. The reason that separate scripts were created was to be able to organize calculations, functions and variables. This prevented the solution from becoming messy.

The solution was built to cover the span of 0 to 20 ampere and -55 to 55 volt. This span will allow the tests to work for the common hardware developed at power solutions.

3.2.1 Output voltage DC-level

The output voltage DC-level test was performed by measuring the maximum value of the output voltage signal with the oscilloscope. (Bryne, S. 2019)

Output voltage DC-level (@ Vin.min)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax
Output voltage DC-level (@ Vin.nom)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax
Output voltage DC-level (@ Vin.max)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax

Figure 3.1: Output voltage DC-level, conditions

Figure 3.1 shows the test conditions for the output voltage DC-level test. The solution was built to perform the test for three voltage levels and five load levels. (Bryne, S. 2020)

3.2.2 Output voltage ripple

The output voltage ripple test was performed by measuring the peak to peak value of the voltage signal on the oscilloscope. (Bryne, S. 2019) (Liu, J. Yue, J. Wang, C. Lyu, F. 2019)

Output voltage ripple (@ Vin.min)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax
Output voltage ripple (@ Vin.nom)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax
Output voltage ripple (@ Vin.max)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax

Figure 3.2: Output voltage ripple, conditions

Figure 3.2 shows the conditions for the output voltage ripple test. The solution was built to perform the test for three voltages and five load levels. If the voltage was nominal and the load level was 0%, 50% or 100%, pictures of the signal were taken. For these three load levels, there was a picture taken for low frequency and one for high frequency. (Bryne, S. 2020)

3.2.3 Dynamic response

The dynamic response test was performed by measuring the minimum and maximum deviation of the output voltage signal with the oscilloscope when the hardware was subject to an electronic pulse. To simulate the electronic pulse, a pulse load was used. The pulse load was sending an electronic pulse from 25% to 75% load with a current slew rate of 2.5A/µs. (Bryne, S. 2019)

Dynamic response, Peak voltage Deviation Positive flank	Dynamic load 25% to 75% 2.5A/µs(slope) @Vin.min
Dynamic response, Peak voltage Deviation Negative flank	Dynamic load 25% to 75% 2.5A/µs(slope) @Vin.min
Dynamic response, Peak voltage Deviation Positive flank	Dynamic load 25% to 75% 2.5A/µs(slope) @Vin.min
Dynamic response, Peak voltage Deviation Negative flank	Dynamic load 25% to 75% 2.5A/µs(slope) @Vin.min
Dynamic response, Peak voltage Deviation Positive flank	Dynamic load 25% to 75% 2.5A/µs(slope) @Vin.min
Dynamic response, Peak voltage Deviation Negative flank	Dynamic load 25% to 75% 2.5A/µs(slope) @Vin.min

Figure 3.3: Dynamic response, conditions

Figure 3.3 shows the conditions for the dynamic response test. The solution was built to perform the test for three voltages. If the voltage was nominal, pictures of the signal were taken. There was a picture taken for both low frequency and high frequency. (Bryne, S. 2020)

3.2.4 Line & load regulation

The line & load regulation test was performed by measuring the minimum and maximum differentiation between the input voltage and the output voltage. (Bryne, S. 2019)

Line & load regulation	@ Vin.min
	@ Vin max

Figure 3.4: Line & load regulation, conditions

Figure 3.4 shows the conditions for the line & load regulation test. The solution was built to use the results of the Output voltage DC-level test to calculate the results for the test. (Bryne, S. 2020)

3.2.5 Output turn on ramp

The output turn on ramp test was performed by measuring the time it took for the printed circuit board to ramp up from zero voltage to the set voltage. (Bryne, S. 2019)

Output turn on ramp	@ Vin.min
Output turn on ramp	@ Vin.nom
Output turn on ramp	@ Vin.max

Figure 3.5: Output turn on ramp, conditions

Figure 3.5 shows the conditions for the output turn on ramp test. The solution was built to perform the test for three voltages. If the voltage was nominal and the load level was 0%, a picture of the signal was taken. (Bryne, S. 2020)

3.2.6 Phase margin, gain margin & cross over frequency

The phase margin, gain margin & cross over frequency test was performed by measuring the phase angle, magnitude and frequency with a Bode diagram generated by a Bode100 instrument. (Bryne, S. 2019)

Phase margin, Gain margin & Cross over frequency (@ Vin.min)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax
Phase margin, Gain margin & Cross over frequency (@ Vin.nom)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax
Phase margin, Gain margin & Cross over frequency (@ Vin.max)	Min load
	25% of Imax
	50% of Imax
	75% of Imax
	100% of Imax

Figure 3.6: Phase margin, gain margin & cross over frequency, conditions

Figure 3.6 shows the conditions for the phase margin, gain margin and cross over frequency test. The solution was built to perform the test for three voltages and five load levels. If the voltage was nominal and the load level was 50%, a picture of the signal was taken. (Bryne, S. 2020)

3.3 Script programming

The structure of the programming was visualized with flowcharts in Microsoft Visio. This made the solution structured and organized. It gave information on how the source code was going to be built. The flowcharts represented the different processes of the automation so that developers could understand it. The scripts were made after the structure of the flow charts. (Clausen, et al. 2013)

The task was to make a solution that the developers could understand, interpret and change. In web development, they split the underlying functions to front end and back end. The front end is how the information is laid out and how the user can interact with it. The back end is focused on the underlying structure of the website, what languages it is written in, the load time and the functions. (Dafforn, E. 2020)

This method was applied to the automatic solution. The processes were placed as subroutines. The subroutines were placed in the script in a logical order. Every subroutine contained calculations and functions to perform the specific process. The use of subroutines gave an easy overview of the script. The front end solution was represented by how the subroutines were named and placed. If you expand the subroutines they open the underlying calculations and functions. This represented the back end of the solution.

When going through the test conditions it was best to use loop functions. Every test parameter was tested under a fixed amount of conditions. When there are fixed values of iterations, it was best to use for-loops. When automating some of the test parameters it was sometimes needed for the oscilloscope to go through a certain amount of sweeps. Then a clear condition was set that the script will continue after a certain amount of sweeps. For these situations, it was best to use while-loops. (McFedries, P. 2001)

The name of the variables played an important role. For the developers to be able to understand and interpret the source code of the solution, the variables were named smart and organized. According to McFedries (2001), you should always give extra thought when naming the variables. If you want an easy-to-follow, easy-to-debug code you should think about your variables.

3.4 Set up

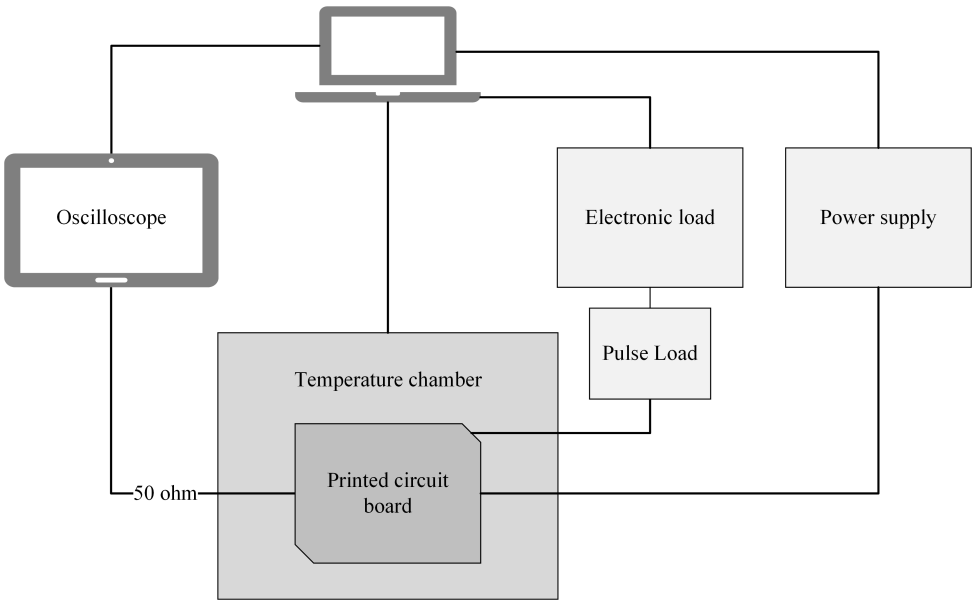


Figure 3.7: Setup for testing

Figure 3.7 shows how the instruments were connected to the computer and the Printed circuit board when building the automatic solution. An oscilloscope, electronic load and power supply were connected to the PCB. The oscilloscope was connected to the PCB through two of its channels. The first channel was connected to the output via a coax cable. The second channel was connected to the positive side of the output via a current probe. With this setup, both the output voltage signal and the output current signal can be measured. The power supply was connected to the input of the PCB. The electronic load was connected to the pulse load which is connected to the output of the PCB. The PCB was placed inside the temperature chamber. All of the instruments were connected to the computer. (Bryne, S. 2019) (Ericsson Internal. 2017)

The idea was that all of the tests would run automatically without any breakpoint. To be able to do that, every instrument had to be connected from the start. The pulse load was only active during the dynamic response test parameter. Even so, it still had to be connected during the other test parameters.

The printed circuit board that was used when building the automatic solution was a test board and not an official product developed by Ericsson. The scripts in EICE are generic and can later be used on all of the printed circuit boards used in radio, baseband and transport networks.

3.4.1 Temperature chamber conditions

Changing the temperature in the temperature chamber was the process that was taking the longest time to set. Because of that, when the temperature was set, all of the test parame-

ters were performed before moving on to the next temperature. So the temperature chamber did not change temperature after each test parameter. This saved time and optimized the automatic solution.

3.5 Evaluation

1. Setup	2. Manual or Automatic	3. Report
Connect instruments to computer and printed circuit board	Perform tests manually or automatic	Summarize results in the verification sheet

Table 3.1: Evaluation process

The solution was tested multiple times with every verification sheet and with the span of 0 to 20 ampere and -55 to 55 volt. To verify that the solution was viable and flexible enough, the scripts were tested on different printed circuit boards and with the use of different instrument models.

While testing and verifying the solution, the time of the processes was measured. When measuring the time, the solution was looked at from an extended viewpoint. Not just looking at the automatic measurement phase, but the entire test process. From setting up the instruments to reporting the results in the verification sheet. The time of every single step of the automation was measured to be able to analyse what could be improved with the scripts.

3.6 Summary of method

The project was carried out on-site at the Power Solution Laboratories. It needed the use of a printed circuit board, electronic instruments, the software EICE and the verification Excel file. The work was done experimentally by programming and developing solutions in EICE. After the automatic solution was built, the solution was analysed and the new testing method evaluated.

Chapter 4

Automatic solution

This chapter explains how the automatic solution is built. The structure of the scripts are presented with Microsoft Visio flow charts.

4.1 Scripts

This section explains each separate script. The structure of each script is presented. The subroutines of each script are explained. The script of the test parameters has subroutines in common. These subroutines are explained in Section 4.2.

4.1.1 User interface

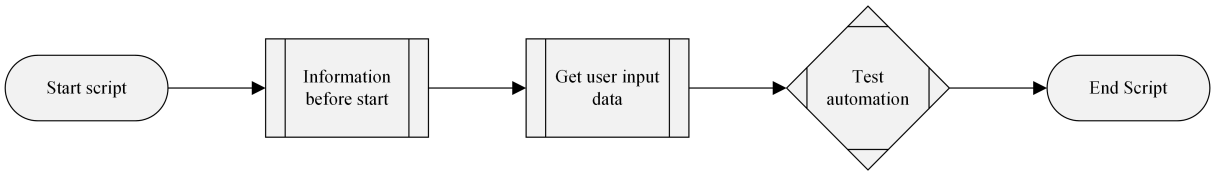
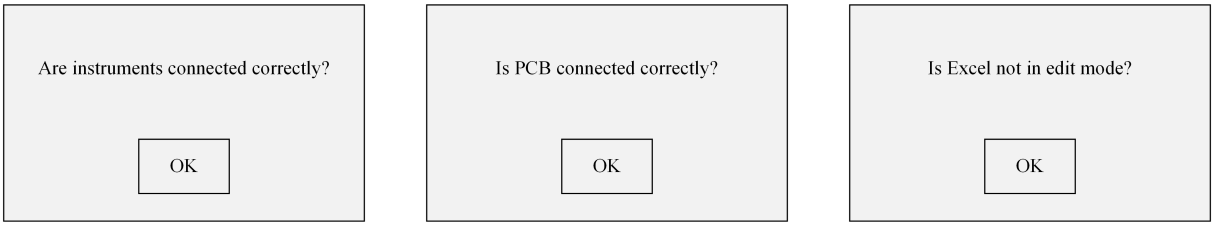


Figure 4.1: User interface, structure

The script that the user opens in EICE is the User interface. Figure 4.1 shows the structure of the User interface script. It comprises the parts the user needs to interact with the program. This is mostly made to make a nice and clean impression. In this way, the user is not thrown directly into the calculations and functions.

4.1.1.1 Subroutine - Information before start

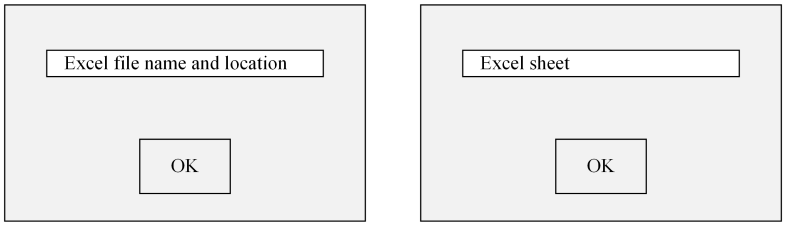


(a) Check instruments (b) Check printed circuit board (c) Excel edit mode

Figure 4.2: Information boxes

The first thing that will appear after starting the script is three questions. Figure 4.2 presents the questions and the look of the question boxes. These will appear one by one. They ask questions to remind the user to double-check the set-up before the automation starts.

4.1.1.2 Subroutine - Get user input data



(a) Excel file name & location (b) Product sheet

Figure 4.3: User input data boxes

To start the automation, the user needs to specify to the program the source of the system requirements. Two input boxes will appear one by one. In the first box, the user inserts the name and location of the verification Excel file. In the second box, the user inserts the name of the product sheets that he/she desires to use. This information is stored in global variables that are used throughout the entire solution. After this process, the user does not have to make any more interactions with EICE. Examples of input:

File: C://desktop/verification.xls
Sheet: Product1_requirements

4.1.2 Test automation

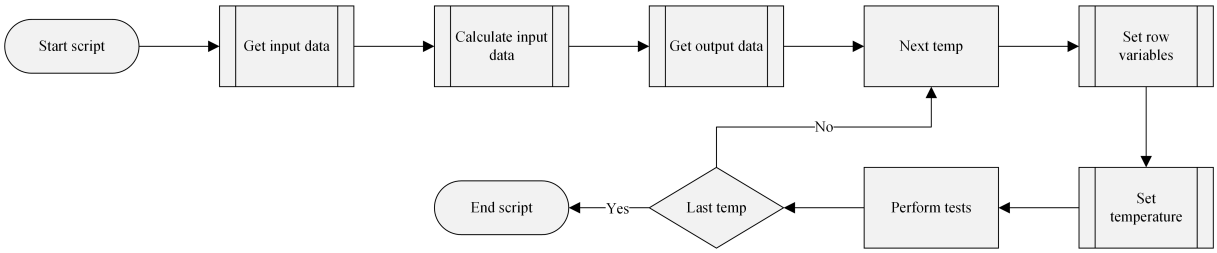


Figure 4.4: Test automation, structure

EICE now got everything to run the automation by itself. It enters the test automation script. Figure 4.4 shows the structure of the Test automation script. It consists of all the processes of the testing.

4.1.2.1 Subroutine - Get input data

The first process of the automation is to gather the input data used to perform the tests. The input data needed is the following:

- Minimum temperature
- Room temperature
- Maximum temperature
- Input voltage
- Input voltage tolerance
- Output voltage
- Nominal current

EICE will execute the external script Find row. This is a function that will find the row of any named string in the verification sheet. This function is explained in greater detail in Section 4.3.1. A loop is set to execute the find row function seven times to get all of the input data.

Measurement at temperature		
-40°C	+25°C	+100°C

Figure 4.5: Temperature input data

The instruments used can only read numerical values. Figure 4.5 shows the temperature input data cells from the verification sheet from Figure 2.5. The temperature levels are listed in a format that is not numerical. To change the format of these cells the script executes the function String manipulation. This function is explained in- depth in Section 4.3.2.

4.1.2.2 Subroutine - Calculate conditions

Min load @ Vin.min
25% of Imax (1,56) @ Vin.min
50% of Imax (3,13) @ Vin.min
75% of Imax (4,69) @ Vin.min
100% of Imax (6,25) @ Vin.min
Min load @ Vin.nom
25% of Imax (1,56) @ Vin.nom
50% of Imax (3,13) @ Vin.nom
75% of Imax (4,69) @ Vin.nom
100% of Imax (6,25) @ Vin.nom
Min load @ Vin.max
25% of Imax (1,56) @ Vin.max
50% of Imax (3,13) @ Vin.max
75% of Imax (4,69) @ Vin.max
100% of Imax (6,25) @ Vin.max

Figure 4.6: Conditions for testing, example

With the input data collected, the test conditions can be calculated. Figure 4.6 specifies the conditions for the test parameter shown for the verification sheet shown in Figure 2.5. It specifies three voltage levels and five load levels. The voltage conditions are minimum voltage, nominal voltage and maximum voltage. To calculate these, the input voltage and the input voltage tolerance are used.

$$V_{tol[V]} = V_{tol[\%]} V_{in} \tag{4.1}$$

$$V_{min} = V_{in} - V_{tol[V]} \tag{4.2}$$

$$V_{nom} = V_{in} \tag{4.3}$$

$$V_{max} = V_{in} + V_{tol[V]} \tag{4.4}$$

The load conditions are min(0%) load, 25% load, 50% load, 75% load and 100% load. In Figure 4.6 the load levels are already calculated. This is to make it simple for the manual tester so that he/she does not have to perform the calculations by hand. (Bryne, S 2020). But it is hard for EICE to pick up these values since they are in the middle of the string. The manipulate string functions could be used to get them, but the calculations are so simple that it is better to do them directly. To calculate these, the input current is used.

$$I_{min} = I_{max} \times 0 \tag{4.5}$$

$$I_{25\%} = I_{max} \times 0.25 \tag{4.6}$$

$$I_{50\%} = I_{max} \times 0.5 \tag{4.7}$$

$$I_{75\%} = I_{max} \times 0.75 \tag{4.8}$$

$$I_{100\%} = I_{max} \times 1 \tag{4.9}$$

When the voltage levels and the load levels are calculated, they are placed in global variables to be used in the test parameters.

4.1.2.3 Subroutine - Get output data

The test parameters are located at different rows depending on which product sheet is used. To be able to place the results at the right places, the script finds where the test parameters are located. The spots for the pictures of the oscilloscope signals are also located. The output data it needs to find are the following:

- Output voltage DC-level, result
- Output voltage ripple, result
- Output voltage ripple, picture
- Dynamic response, result
- Dynamic response, picture
- Line & load regulation, result
- Output turn on ramp, result
- Output turn on ramp, picture
- Phase margin, gain margin & cross over frequency, result
- Phase margin, gain margin & cross over frequency, picture

To do this, EICE executes the external script "Find row". This function is explained in greater detail in Section 4.3.1. A loop is set to execute the find row function ten times to get all of the output data. These locations of the tests are placed in global variables so they can be used in the tests.

4.1.2.4 Process - Loop for temperatures

The script loops for the temperature levels. The loop runs for three laps going through the minimum temperature, room temperature and maximum temperature. It starts with the minimum temperature and ends with the maximum temperature. As explained in Section 3.4.1 all of the tests are performed before changing temperature.

4.1.2.5 Subroutine - Set row variables

Inside the temperature loop, the script is placing the location of the outputs into global variables. This is because the output locations have to reset for each temperature.

4.1.2.6 Subroutine - Set temperature

The script sets the temperature to the input data temperature levels depending on which lap in the temperature loop is active. The temperature chamber is initiated and adjusted to that temperature. A condition is set so that the script will continue once the temperature has reached within one degree of its set value. When the temperature is within that interval specified in Equation 4.10, the script continues to run.

$$T_{set} - 1°C > T_{set} > T_{set} + 1°C \tag{4.10}$$

4.1.2.7 Subroutine - Perform tests

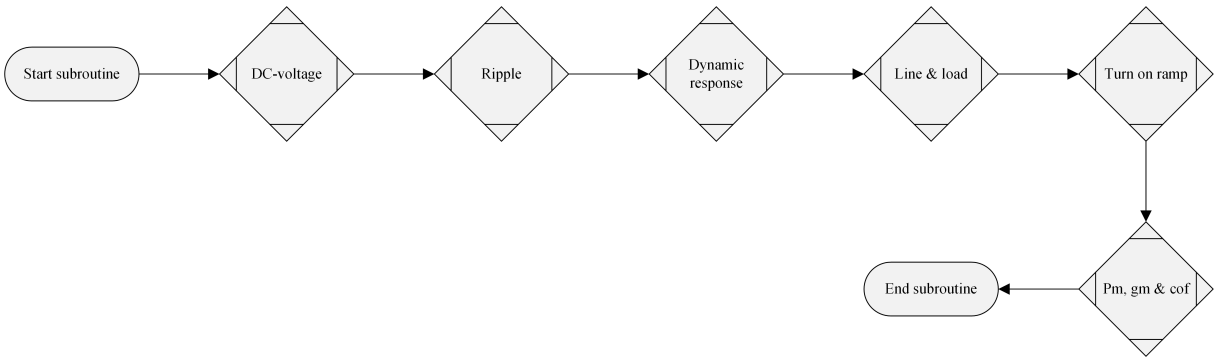


Figure 4.7: Perform tests

EICE has now performed every pre-set needed before entering the test scripts. Figure 4.7 shows the structure of the subroutine Perform tests. This process consists of the six test parameters. The test parameters are placed as external scripts in the order of the verification sheets.

4.1.3 Output voltage DC-level

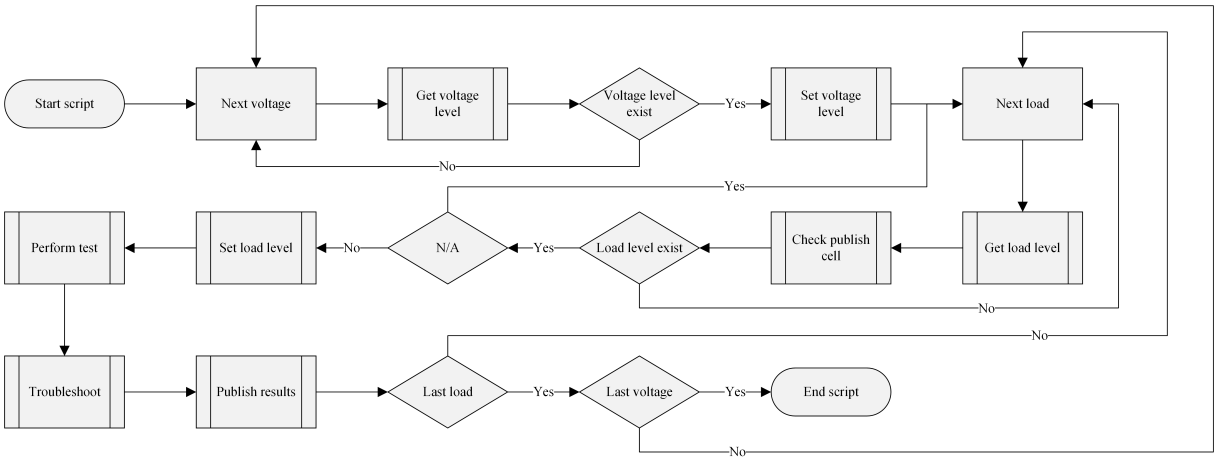


Figure 4.8: Output voltage DC-level, Structure

Figure 4.10 shows the structure of the output voltage dc-level script.

4.1.3.1 Subroutine - Perform test

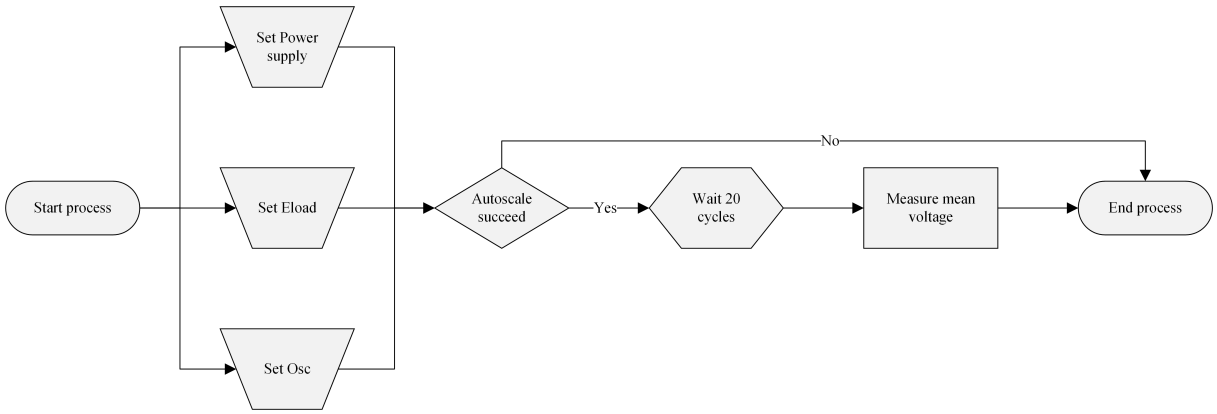


Figure 4.9: Output voltage DC-level, Perform test

Figure 4.9 shows the structure of the perform test subroutine for the output voltage dc-level test. The first step is to initiate the power supply and electronic load. The power supply is set to the voltage given by the voltage loop. The electric load is set to the load given from the load loop. The oscilloscope is set to DC-mode. It is set to autoscale the signal. To get a good measurement the script waits for 20 sweeps. After that, the oscilloscope measures the mean voltage.

4.1.3.2 Subroutine - Publish results

The result of the test is placed in the correct cell.

4.1.4 Output voltage ripple

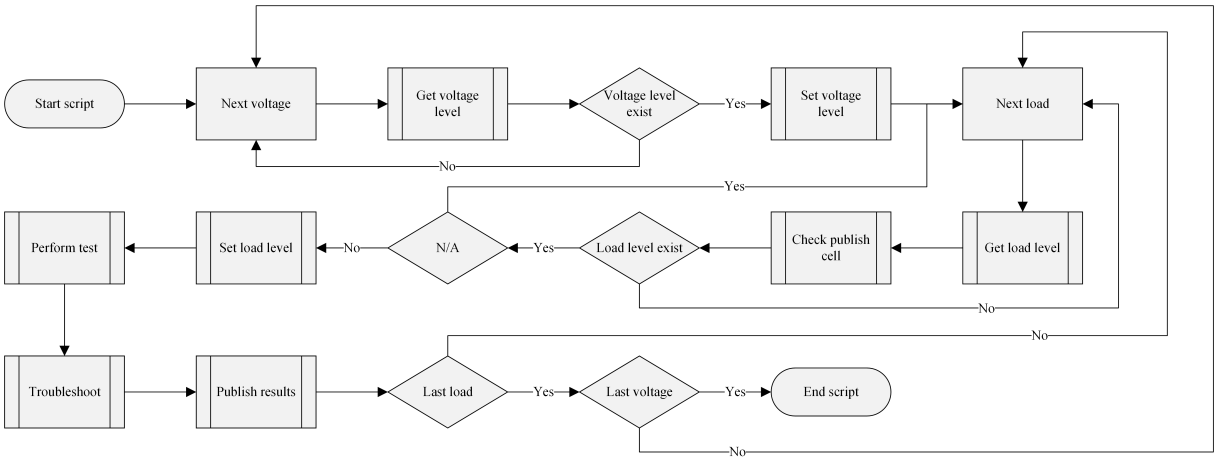


Figure 4.10: Output voltage ripple, Structure

Figure 4.10 shows the structure for the output voltage ripple script.

4.1.4.1 Subroutine - Perform test

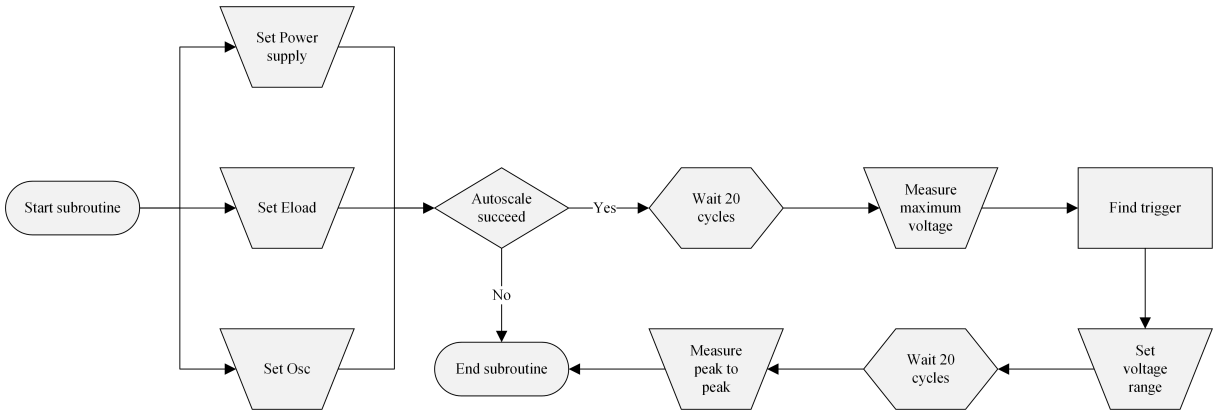
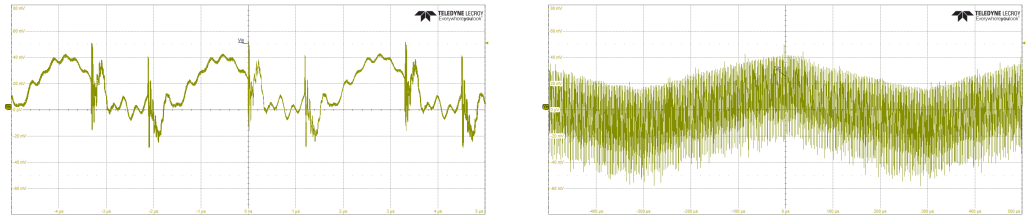


Figure 4.11: Output voltage ripple, Perform test

Figure 4.11 shows the structure of the perform test subroutine for the output voltage ripple test. The first step is to initiate the power supply and electronic load. The power supply is set to the voltage given by the voltage loop. The electric load is set to the load given from the load loop. The oscilloscope is set to AC-mode. It is set to autoscale the signal. If the auto-scaling succeeds it waits for 20 cycles and measures the maximum voltage. The oscilloscope sets a trigger point slightly above the maximum voltage. The oscilloscope is set to normal scale. The trigger level ticks down until the signal starts to trigger. This gives a single signal of the maximum voltage value. With this voltage value, the oscilloscope scales the signal. Then it is set to wait for 20 sweeps to get a good measurement. At last, it measures the maximum peak to peak value of the signal.

4.1.4.2 Subroutine - Publish results



(a) Low frequency (b) High frequency

Figure 4.12: Output voltage ripple, signal

The result of the test is placed in the correct cell. When the conditions for taking pictures are true, the signal is first set to low frequency and the oscilloscope captures a picture. Then it is set to high frequency and the oscilloscope captures a picture. When the pictures are taken they are placed in the correct cells. Figure 5.2 shows the captures of the oscilloscope signals in low frequency and high frequency.

4.1.5 Dynamic response

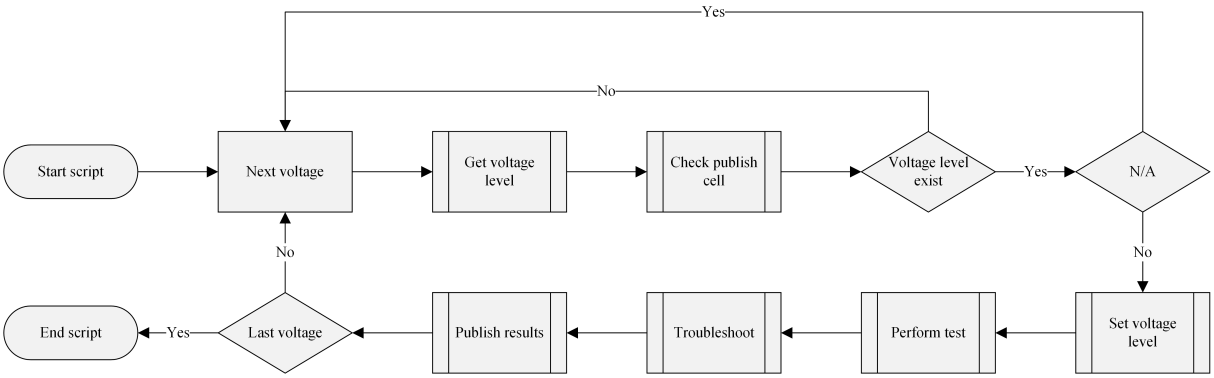


Figure 4.13: Dynamic response, structure

Figure 4.10 shows the structure for the dynamic response script.

4.1.5.1 Subroutine - Perform test

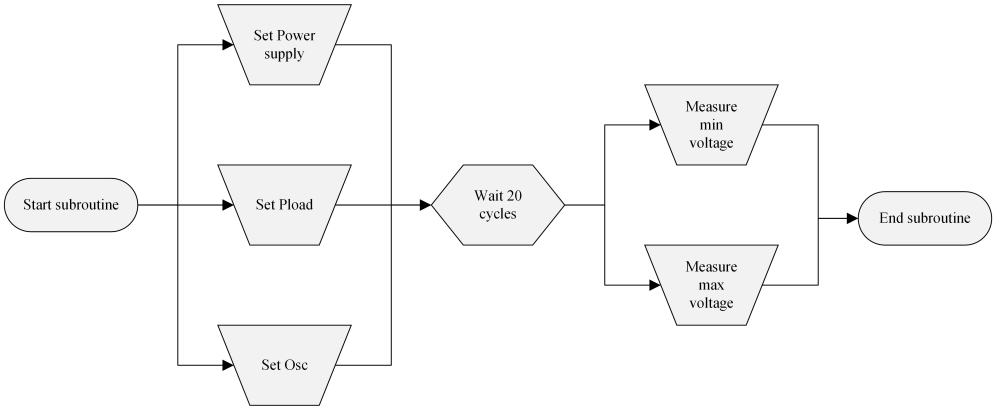
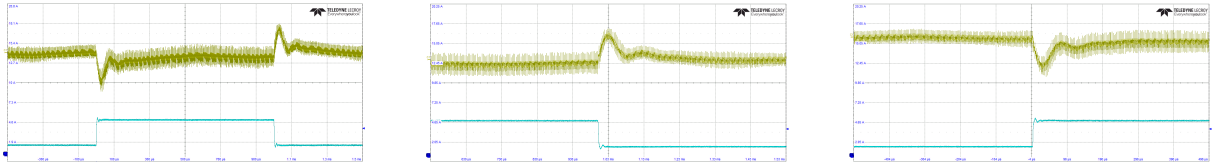


Figure 4.14: Dynamic response, perform test

Figure 4.14 shows the structure of the perform test subroutine for the dynamic response test. The first step is to initiate the power supply and pulse load. The power supply is set to the voltage given by the lap of the voltage loop. The pulse load is set to deliver a pulse. The oscilloscope is set to include both the output voltage signal and the current signal on the screen. The script will wait for 20 sweeps to get a good measurement. Then the oscilloscope will measure the minimum voltage and maximum voltage that occurs because of the electric pulse.

4.1.5.2 Subroutine - Publish results



(a) Pulse (b) Falling edge (c) Rising edge

Figure 4.15: Dynamic response, signal

The result from the measurement is placed in the correct cell. When the conditions for taking pictures are true, the oscilloscope first captures a picture of one entire pulse. Then it captures a picture of the falling edge of the pulse. At last, it captures a picture of the rising edge of the pulse. Figure 4.15 shows the captures of these oscilloscope signals.

4.1.6 Line & load regulation

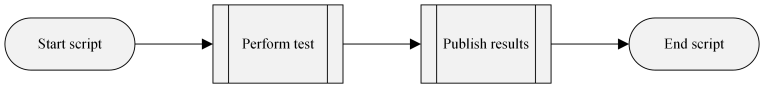


Figure 4.16: Line & load regulation, structure

Figure 4.16 shows the structure for the line & load regulation script. Line & load regulation only has a perform test subroutine and a publish result subroutine. This is because the test is based on the result from the output voltage DC-level test and does not have any own conditions.

4.1.6.1 Subroutine - Perform test

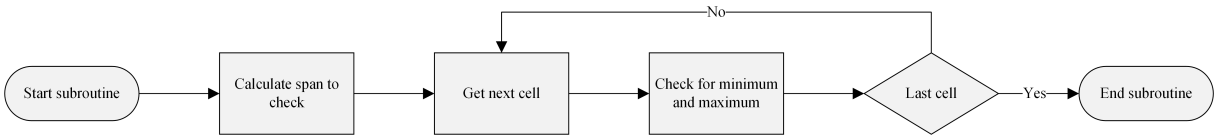


Figure 4.17: Line & load regulation, perform test

Figure 4.17 shows the structure of the perform test subroutine for the line & load regulation. The script starts by calculating the span of cells from the output voltage DC-level test. This is the span of all the results it has to check through. A loop is set to check the result in each cell and compare it to the result from the previous cell. When the cells are checked, the differentiation of the minimum/maximum value and the input voltage is calculated.

4.1.6.2 Subroutine - Publish results

The results are placed in the correct cell.

4.1.7 Output turn on ramp

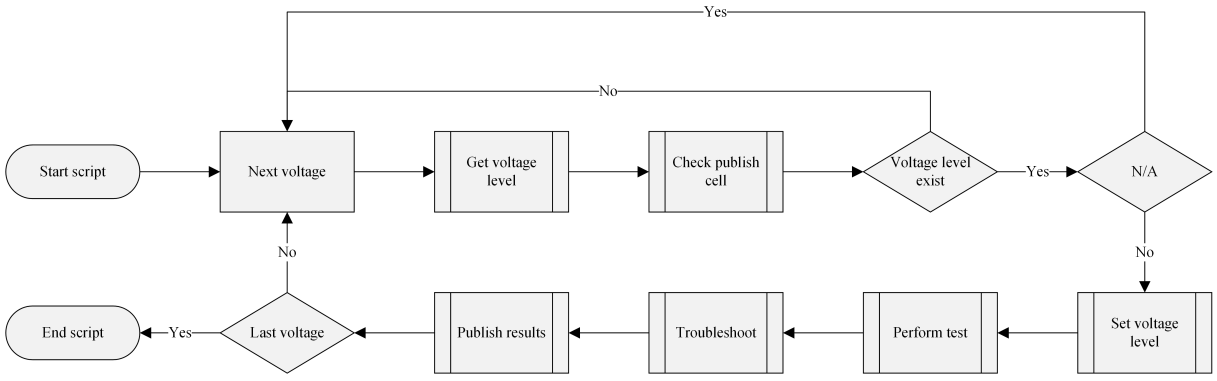


Figure 4.18: Output turn on ramp, structure

Figure 4.18 shows the structure for the output turn on ramp script.

4.1.7.1 Subroutine - Perform test

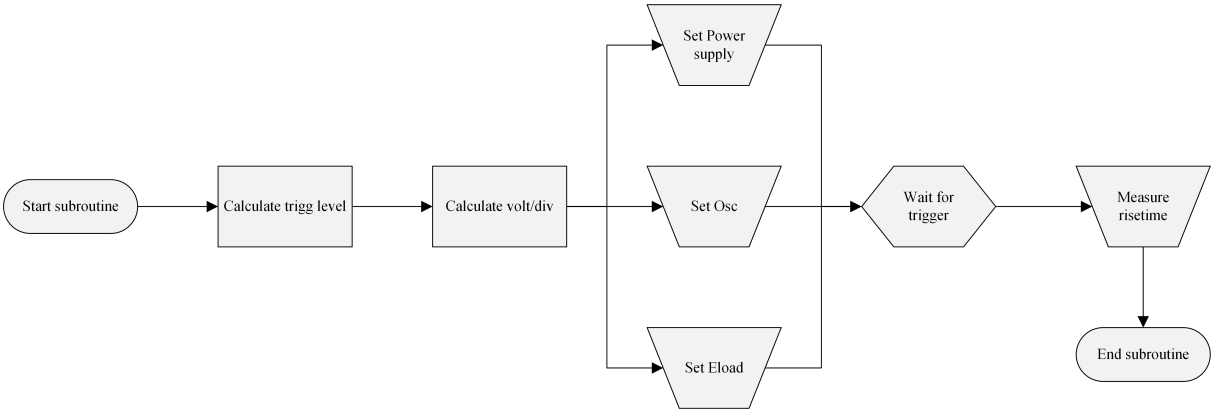


Figure 4.19: Output turn on ramp, perform test

Figure 4.19 shows the structure of the perform test subroutine for the output turn on ramp test. The first step is to calculate the trigger level. The trigger level is set to 75% of the output voltage to be able to find the slope. Then the script is initiating the power supply and the electronic load. It sets the oscilloscope to capture one single trigger of the start ramp of the output voltage. Then the oscilloscope measures the time of the ramp startup.

4.1.7.2 Subroutine - Publish results

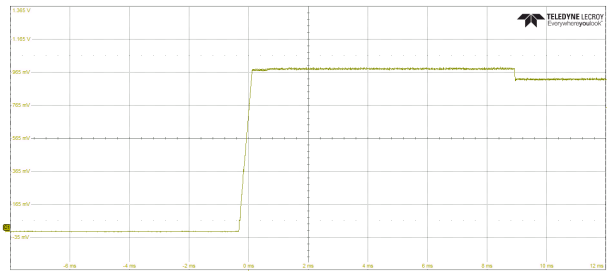


Figure 4.20: Output turn on ramp, signal

The result from the measurement is placed in the correct cell. When the conditions for taking pictures are true, the oscilloscope captures a picture of the output turn on ramp. Figure 4.20 shows the captures of these oscilloscope signals.

4.1.8 Phase margin, gain margin & cross over frequency

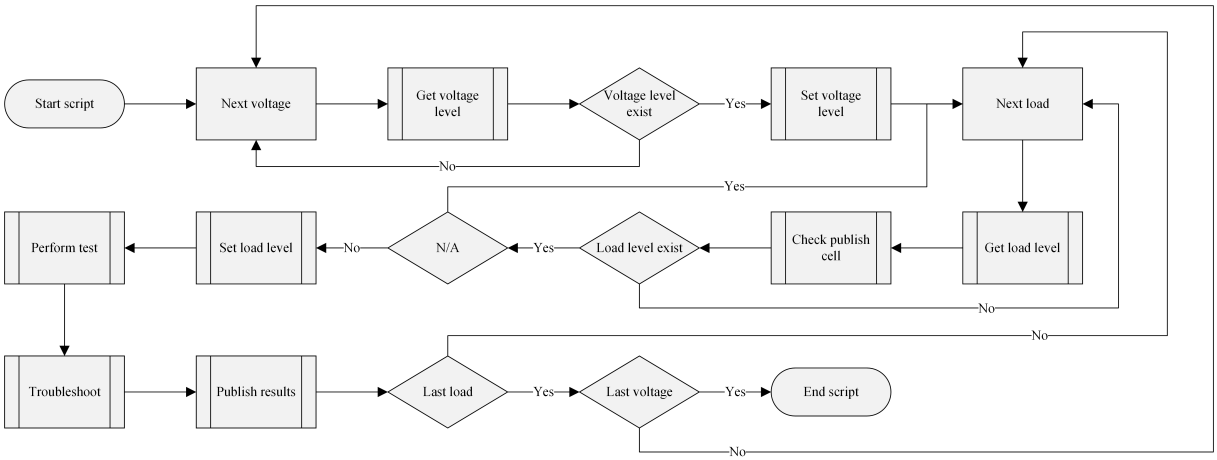


Figure 4.21: Phase margin, gain margin & cross over frequency, structure

Figure 4.21 shows the structure for the phase margin, gain margin, cross over frequency script.

4.1.8.1 Subroutine - Perform test

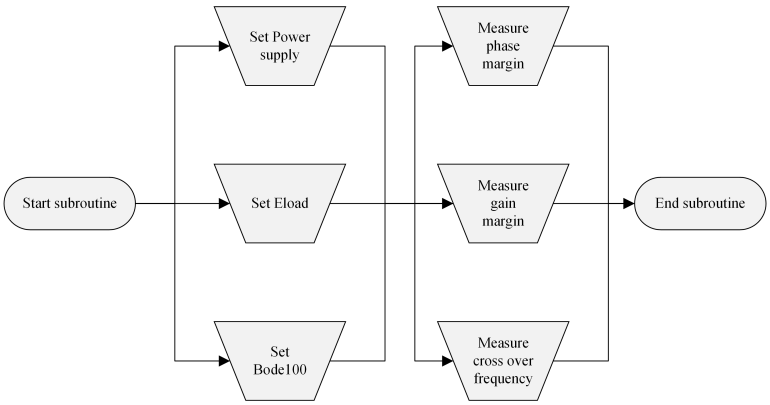


Figure 4.22: Phase margin, gain margin & cross over frequency, perform test

Figure 4.22 shows the structure of the perform test subroutine for the phase margin, gain margin and cross over frequency test. The first step is to initiate the power supply and electronic load. The power supply is set to the voltage given from the voltage loop. The electric load is set to the load given from the load loop. The oscilloscope does not need to perform any measurement on this test parameter. The measurements are performed by the Bode100 instrument. It is initiated to the general settings and performs a sweep. EICE measures the phase angle, magnitude and frequency from the Bode100 plot.

4.1.8.2 Subroutine - Publish results

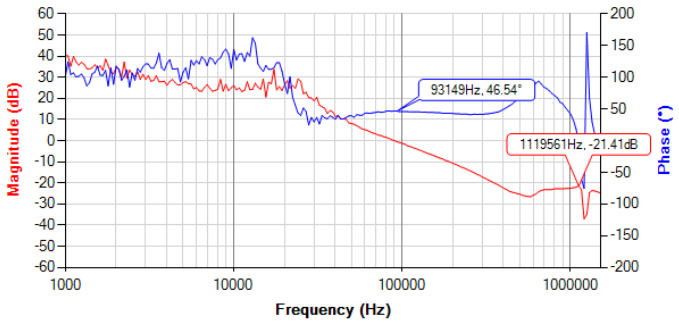


Figure 4.23: Phase margin, gain margin & cross over frequency, signal

The result from the measurement is placed in the correct cell. When the conditions for taking pictures are true, the Bode100 captures a picture of the plot. Figure 4.23 shows the capture of the Bode100 plot.

4.2 Common subroutines in the test parameter scripts

4.2.1 Subroutine - Get voltage level

The script enters the verification sheet to check which voltage level is active. This is done by reading the information from the test condition cell. The script checks if the active voltage level is true for the product sheet in use. This is done because, in the Sau requirement verification sheet, there is only a condition to test with nominal voltage. The minimum and maximum voltage do not exist. When that happens the script will move to the next lap and check the next voltage.

4.2.2 Subroutine - Set voltage level

The active voltage level is set into a variable.

4.2.3 Subroutine - Get load level

The script will now enter the excel sheet to check which load level is active. It does this by reading the information from the condition cell.

25% of Imax (1,56) @ Vin.min

Figure 4.24: One test condition, example

Figure 4.24 displays the load level calculated in Ampere. For every product sheet, this ampere value is unique. This is because the verification sheets have different input current.

To make the solution generic EICE only reads the load level in percentage. This is done by executing the function string manipulation. This function is explained in-depth in Section 4.3.2.

The script checks if the active load level is true for the product sheet in use. This is done because, in the Primary requirement-verification sheet, the conditions for 25% load and 75% load does not exist. When that happens the script will move to the next lap and check the next load.

4.2.4 Subroutine - Check publish cell value

The script checks the value in the cell where the result will be placed. This is to check if the condition is N/A(not applicable). If the cell contains the string N/A, the test will not be performed.

4.2.5 Subroutine - Set load level

The load level is set into a variable.

4.2.6 Subroutine - Troubleshoot

The troubleshooting process checks if any errors have occurred when performing the test. This function is explained in-depth in Section 4.3.3.

4.3 Functions

4.3.1 Find row

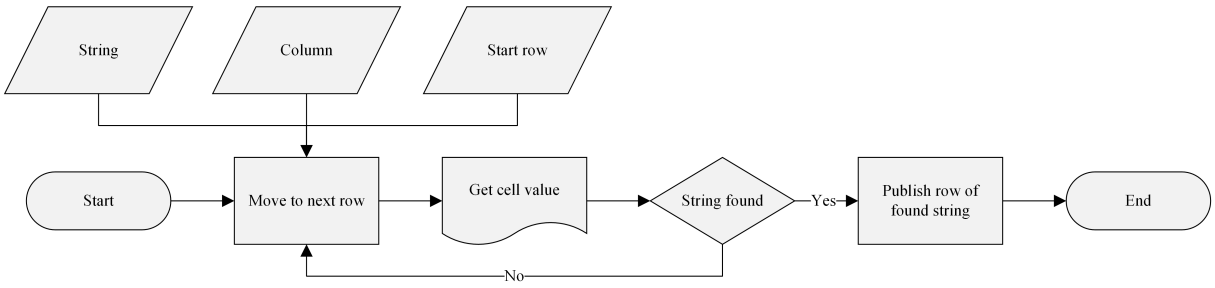


Figure 4.25: Find row function

The input data and output data of the four verification sheets are located in the same columns but at different rows. To make a general solution, EICE needs to locate the rows to be able to get that data. This is solved with the function Find row. It is made as an own separate script. What the function does is that it searches a specific column in the excel document for a specific string, and returns the row value of the cell where that string was found. To do this it needs three global input data values.

- String
- Column
- Start row

The string specifies what string to look for. The column specifies the column which is going to be searched. The start row specifies at what row the search will start. The start row exists because when finding some of the data it is unnecessary to start the search at row one. If it is known that the output string is for certain placed below a specific row, then it saves time by starting the search further down. When the string is found it will send the row of that cell to a global variable. When the row is found it is very easy to gather the information. The script can locate all cells linked to that row and use them to get data or write data.

4.3.2 String manipulation

There is no way to manipulate, change or read parts of a string in EICE. The software does not have any function to perform this. Since EICE can read and write cells in Excel, it can let Excel do the string handling.

String	Delimiter	Counter	Result
+50℃	°	4	+50

Figure 4.26: String manipulation, function

The function is located on a separate sheet in the verification Excel file. It needs two input data parameters from EICE. A string and a delimiter. The string is the string that needs to be manipulated. The delimiter is the sign of where the string should be split. EICE will write the string in the cells of the function. The counter will count all the elements up to the delimiter. The resulting cell will print all the elements counted from the left of the string. EICE can now read the result cell and get the answer needed.

4.3.3 Terminate script

The terminate script is a function that ends the script. When it is called, it will immediately turn off all the instruments. It will also display a message containing what error caused the function to be executed.

4.3.3.1 Auto scale fail

The most common error that occurs is happening when the oscilloscope is auto-scaling the signal. Sometimes the trigger is not set in the right place and the oscilloscope gets stuck. Even when giving the oscilloscope some time to initiate the setup. The reason for this is unknown.

To approach this in EICE there are conditions placed in the scripts whenever the oscilloscope is auto-scaling. After the auto-scaling is performed, the script will check if the oscilloscope is triggering or not triggering. If it is triggering, the test will continue. If it is not triggering, the test will restart. If the auto-scaling would fail twice in a row, the Terminate script will be executed. There is a low chance of the autoscale to fail twice in a row. If this would happen, there might be something wrong with the setup. Therefore it is safer to turn them off.

4.4 Variable names

The names of the variables are split into three classes and one sub-class. The classes are the following:

- Global
- Local
- Sub

The subclass is the following:

- Local

Global is for global variables. Local is for local variables. Sub is for variables used within the subroutines. The sub-class is for the variables used for loop indexes. The classes and sub-class start with capital letters. Every other process is named with small letters. An underscore is placed between classes, subclass and words. Two examples:

- Sub_Loop_currents
- Global_voltage_level

4.5 Summary of automatic solution

The automatic solution is automating six test parameters used for verification. It is structurally built and well organized with the use of external scripts and subroutines. With the use of Microsoft Visio flow charts the scripts were optimized to be as efficient as possible. There are three main functions created that are used throughout the entire solution. These functions made it possible to build the concept of a general automatic solution used for verification.

Chapter 5

Evaluation

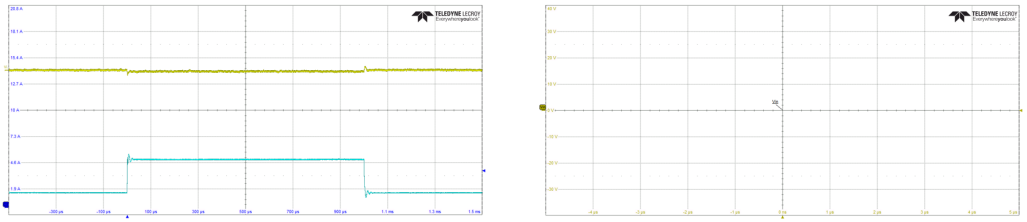
This chapter explains the evaluation process.

5.1 Verifying solution

The solution was tested multiple times with the use of every verification sheet. During the tests, a few smaller problems were found that prevented the solution from working properly. These were fixed by making some tweaks and changes to the scripts. The solution was also tested with different instrument models and different printed circuit boards. After running the tests, two issues occurred that prevented the solution from working. This occurred in the test parameters dynamic response and phase margin, gain margin & cross over frequency. Other than this, the solution opposes no problems for any of the instruments and works without any problem for the different printed circuit boards.

5.2 Volt/div limitation

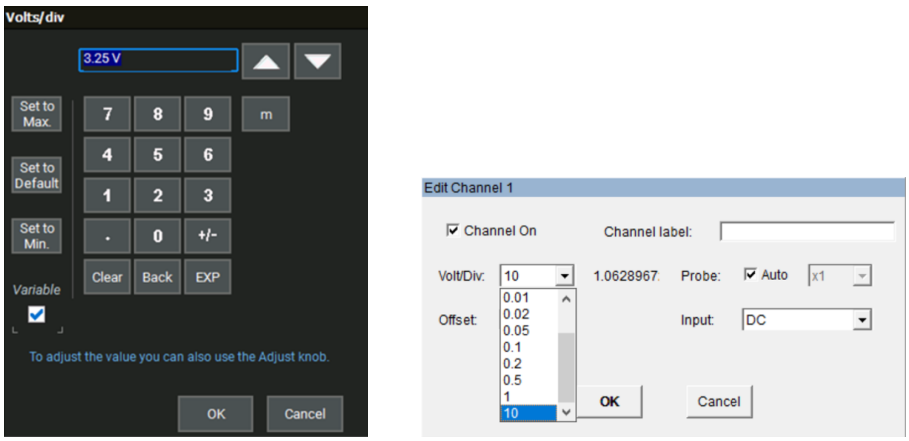
When testing dynamic response there is a need to scale the signal on the oscilloscope for the three different captures shown in Figure 4.15. When scaling the signal, the Volt/Div must be set so that the signal is placed within the boundaries of the oscilloscope screen. If the signal would appear outside the boundaries, it will not be able to measure.



(a) Correctly scaled signals (b) Faulty scaled signals

Figure 5.1: Scaling signal on the oscilloscope

Figure 5.1a shows a correctly scaled signal for the dynamic response test parameter. Figure 5.1b shows an oscilloscope screen where the signal is placed outside the boundaries of the screen. It must be possible to set any value as volt/div if the solution is going to be general. The limiting factor is that on the oscilloscope panel in EICE, there is a drop-down list of numbers to choose from.



(a) On the Oscilloscope (b) In EICE

Figure 5.2: Volt/div setting

Figure 5.2a shows the Volt/Div setting on the oscilloscope. On the oscilloscope, any integer can be set. The values can be set with the precision of two decimals. Figure 5.2b shows the Volt/Div setting from EICE. Only these numbers in the list can be set. Other numbers are not possible to set from EICE. The numbers in the drop-down list are the following:

0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 10

There is one way to make it possible to set a few more numbers from EICE. This is if the probe is scaled. Then the numbers could be multiplied by these scale-values:

x1, x2, x5, x10, x20, x50, x100

Scaling the probe would not only scale the number but also the value of the current. This would cause trouble in other parts of the scripts. It would also require a complex algorithm to utilize all of the numbers. Even if probe scaling was implemented, there would still be many numbers that could never be set. Prime numbers are impossible to set since they cannot come from any product of any of the numbers on the list. What EICE does when it is told to set a number that does not exist in the drop-down list is that it rounds up to the closest value. If for example number 3 is set it will scale the volt/div setting to 10. This makes it so that the signal might appear outside of the boundaries of the oscilloscope screen. The large gap between values 1 and 10 makes up a big dead zone with many values that can not be set.

5.3 Bode100 bug

When testing the phase margin, gain margin & cross over frequency there is a need to get measurements from the Bode100 plot. When making a general solution the measurements need to be placed into variables. A bug was found with the Bode100 instrument panel that made EICE crash every time measurements from the Bode plot were set into variables. This bug was fixed directly by the software developer of EICE.

5.4 Script properties bug

When the solution was tested from another developer's computer/user an error occurred. When the developer opened the script in EICE an unusual property of the script changed. The picture format changed from tiff to pdf which made it not possible to publish pictures from the oscilloscope. This bug was fixed directly by the software developer of EICE.

5.5 Time measurement

Steps	**Time**
1. Set up	15 min
2. Find input and output data	20 min
3. Change the temperature to low	21 min
4. Run test parameters	23 min
5. Change the temperature to room	12 min
6. Run test parameters	23 min
7. Change the temperature to high	14 min
8. Run test parameters	23 min
Summary	**2 h 30 min**

Table 5.1: Time of automatic test processes

Table 5.1 shows how long time it takes for every step in the automatic testing process and the final summary. The time it takes to set up the instruments and settings in EICE is around 15 minutes. This data is just an approximation. Measuring the time of the setup is inconsistent since it varies from person to person. For EICE to find the input and output data in the verification sheet takes around 20 minutes. This factor varies a slight amount for each verification sheet since data is located at different places. Changing the temperature in the temperature chamber is taking up a third of the time of the entire process. To run the test parameters takes around 23 minutes each time. This varies for each verification sheet since some have more or less test conditions. The entire process takes around two and a half hours to complete.

Test parameters	**Time**
1. Output voltage DC-Level	5 min
2. Output voltage ripple	13 min
3. Dynamic response	1,5 min
4. Line & load regulation	0,5 min
5. Output turn on ramp	1 min
6. Phase margin, gain margin & cross over frequency	7 min

Table 5.2: Time of separate test parameters

Table 5.2 shows the results for how long time it takes to perform each test parameter automatically. The test parameter line and load regulation takes the shortest amount of time. That is because it is based on the result of another test parameter and does not have any own test conditions. The test parameter output voltage ripple takes the longest amount of time. That is because it contains the most amount of steps to be performed. One thing noticed is that the function "String manipulation" (Section 4.3.2) is slow. It takes time for Excel to process the string and for EICE to communicate back and forth. The function is used many times throughout the entire solution and is therefore making it quite slow.

5.6 Summary of evaluation

The solution was tested multiple times with the use of every verification sheet, different instrument models and different printed circuit boards. An error occurred with the volt/div setting in the test parameter dynamic response. This error prevents the solution from working properly. Two bugs were found in EICE. These were fixed directly by the software developer. The time it takes to run the entire process is about two and a half hours. To run the tests takes 23 minutes for each temperature. The test parameter output voltage ripple takes the longest amount of time. One process that is causing the solution to run slowly is the "String Manipulation" function.

Chapter 6

Conclusion

Is the current version of EICE capable of creating a general automatic testing method that is functional?
The limitation with the volt/div setting in EICE is preventing the solution from working properly. This problem can not be fixed by changing the scripts. The only way to fix it is to make a software update to EICE. A suggestion to solve this issue would be to implement that any numerical value can be set to the volt/div section. This is possible on the physical panel on the oscilloscope, so why not in EICE as well. Currently, the error is only affecting the test parameter dynamic response, but it might also affect other test parameters created in the future. If this is fixed, it is certainly possible for this type of testing method to work.

When the time was measured it was found that the function "String Manipulation" was slow. A suggestion to solve this issue would be to develop a function in EICE that can change, read parts of and manipulate strings. This would require a software update to EICE. It would make the solution faster and less messy. It would no longer require Excel to perform calculations. It is always better to let the software adapt to the documents, not the other way around. If this type of testing method is going to be used, it is suggested that this is implemented.

How much time is saved by performing the tests automatically?
The time it takes to run the entire solution is around two and a half hours. Since testing a rail manually can take up to eight hours, the automation is more than three times as fast. It requires no effort compared to performing the tests manually. Considering this, the automation is a great method for saving time. The developers can do other important tasks while the tests run by themselves. The results can then be reviewed after. The method is also standardized which means that it always takes the same amount of time to run.

6.1 Negatives of automatic testing

When performing the tests manually there is a chance that the developer encounter deviations with the measurements. When manually adjusting the instrument panels on the oscilloscope

the developer might notice deviations with the signals that would not be recognized by EICE. EICE only looks at the intervals specified in the scripts and would not recognize any odd signal behaviour outside that. To create an automatic solution that checks for deviations in the signals an AI system must be created that learns the behaviour and error probabilities of the signals.

6.2 Future work

Since the solution is built around the verification Excel file, it is dependent on where data is located in the sheets. Therefore it might affect the scripts if rows and columns would change in the future. If this type of testing method is going to be implemented, the scripts have to be edited whenever the verification Excel file gets changed.

A method that could be used to get a better estimate of the time saved would be to also perform manual testing and measure the time. The time of the two methods could then be compared to each other. But it is hard to measure the time of manual testing because of the inconsistency. Every developer performs the tests manually at their own pace. There are many factors of uncertainty such that every person has a different experience with testing and therefore it will take a longer or shorter time.

To verify the quality of the testing, the methods for performing the test parameters automatically could be validated. Currently, the measurements are approximated with a certain amount of sweeps and iterations. To improve the quality it should be studied how to more accurately perform the measurements with EICE.

To finish the testing method would be to develop the remaining test parameters. The solution could also be optimized to make it faster and more secure. Here are some examples of ideas that could be relevant:

- Make the solution work for a wider span of voltage- and ampere levels.
- Implement more error handling. An error condition could be set at every place in the script where there is a risk for instrument failure.
- Some of the test parameters have the same script structure. These could be optimized by putting them together.
- Set the oscilloscope channels as variables. This would make the setup less limited.
- Make the scripts function for 8-channel oscilloscopes.
- Send emails from EICE with the progress of the testing. That way the user can get notifications on their cellphone without monitoring the test rig.

List of references

Bryne, S. (2019-10-14). [Microsoft Word-document] *Verification instruction for board power*. Ericsson Internal. ericsson.sharepoint.com

Bryne, S. (2020-12-18). [Microsoft Excel-document] *DS VEM VR template for Power Solution*. Ericsson Internal. ericsson.sharepoint.com

Clausen, L.R.B. Conradt, M. (2013). *System to visualize additional information on source code* (United States Patent: 9274756). United States Patent

Dafforn, E. (2020). *Information Technology (IT) Professionals: A Practical Career Guide*, Rowman & Littlefield

Ericsson. (2017-03-23). [PowerPoint-presentation] *Introduction to EICE and autotest pa02*. Ericsson Internal. ericsson.sharepoint.com

Ericsson. (n.d.). [PowerPoint-presentation] *Verification process*. Ericsson Internal. ericsson.sharepoint.com

Johnsson, G. (2020). *EICE* [Computer software]. GJC Test & Software

Liu, J., Yue, J., Wang, L., Wu, C., Lyu, F. (2019). *A Low-Cost, High-Precision Method for Ripple Voltage Measurement Using a DAC and Comparators*, Electronics

McFedries, P. (2001) *Special Edition, Using Javascript*, Que

Rosén, L. Sahar, S. (2012). *Small Electronic Load Design and analysis of a small electronic load for testing on-board DC/DC converters*, Chalmers University of Technology, Department of Electrical Engineering

Örnberg, E. (2015-06). [Microsoft Word-document] *Installation guide and manual for EICE - Ericsson Instrument Control Environment version 1.0.3*. Ericsson Internal. ericsson.sharepoint.com

Appendices

Appendix A

List of instruments and printed circuit boards

This list contains all of the instruments and printed circuit boards used when building the automatic solution. All of the instruments and printed circuit boards used when testing the solution are not included.

Temperature Chamber
Vötsch
VT4002 climate chamber

Oscilloscope
Teledyne LeCroy
WaveSurfer 4034HD

Electronic Load
Chroma
DC Electronic Load Mainframe

Power Supply
Delta Elektronika
800 Watts DC Power Supply, SM-70-24

Pulse Load
Custom made
Ericsson (Rosén, L. Sahar, S. 2012)

Bode100
OMNICRON Lab
Vector Network Analyzer - Bode100

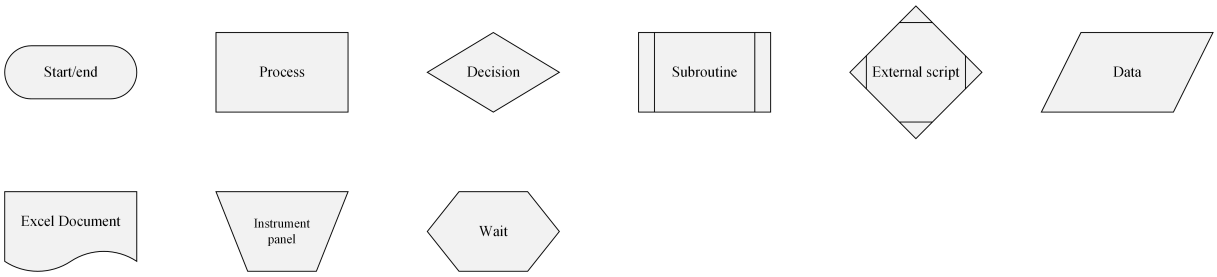
Probe
Teledyne LeCroy
AP015 Current Probe

Printed Circuit Board 1
Test board
Input voltage 55 V, Output voltage 5 V

Printed Circuit Board 2
Phoenix
Input voltage 48 V, Output voltage 1 V

Appendix B

List of Visio elements



DEPARTMENT OF ELECTRICAL ENGINEERING
EENX20
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

CHALMERS