

Digitizing and Modernizing a HP141 Display



Prepared by:
Bonga Njamela

Prepared for:
Dr Stephen Paine
Department of Electrical Engineering
University of Cape Town¹

May 23, 2025

¹Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for the qualification of a Bachelor of Science in Electrical and Computer Engineering.

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

A handwritten signature in black ink, appearing to read "Afambe".

May 23, 2025

Name Surname

Date

Acknowledgements

When you ask God for a gift, be thankful if he sends not diamonds, pearls, or riches but the love of real, true friends.

—*Muhammad Ali*

To my supervisor, Dr Stephen Paine, thank you for pushing me to reach my full potential and completing this research project. I am truly grateful for showing me what it means to be a good engineer.

To Michael Inggs, thank you for providing the necessary information for starting the project. The technical data, notes and videos that you sent to me at the beginning of the project were a lamp on the path to completing the project.

To my teachers who shared their knowledge along my academic journey, thank you for the great work that you continue to do in opening doors to a brighter future.

Thank you to my friends Thato Makhubedu and Ditiro Nkuna for their help when I did not have a place to stay while writing this report. Thank you to Steve Jermy, whose advice allowed me to remain in the city long enough to complete the project.

To the Almighty God and my mother, thank you for applying the patience that was required to mould me into the person that I am today.

Abstract

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Contents

List of Figures	viii
Abbreviations	xi
1 Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Project Requirements	2
1.4 Scope & Limitations	4
1.5 Report Outline	4
2 Literature Review	5
2.1 Mitigating Spectrum Analyzer Obsolescence: A Historical Perspective	5
2.1.1 Introduction to Spectrum Analysis	5
2.1.2 Early Developments in Waveform Analysis	6
2.1.3 Advancements in Spectrum Analysis Technology	6
2.1.4 Early Modifications to Spectrum Analyzers (1960s - 1980s)	7
2.1.5 Commercial Digitization of Spectrum Analyzers	8
2.1.6 Obsolescence Mitigation and Modernization Strategies	8
2.1.7 Conclusion	9
2.2 A Preview of the Operation of Spectrum Analyzers to Establish the Context for Digitization and Modernization	10
2.2.1 Classifications of Spectrum Analyzers	11
2.2.2 Primary Components of Spectrum Analyzers	11
2.2.3 Features of Modern Spectrum Analyzer Displays	11
2.3 Digitizing Spectrum Analyzer Outputs	11
2.3.1 Output Voltage Regulation and Preparation for Frequency Analysis	11
2.3.2 Transforming Spectrum Analyzer Output Signals to Digital Frequency Domain	11
2.3.3 Interfacing Computers with Spectrum Analyzers	11
3 Methodology	12
3.1 Methodology Outline	12
3.2 Phases in the Design Process	13
3.3 Requirements Review	14
3.3.1 Comparing the HP141T System to Modern Spectrum Analyzers	14
3.3.2 Data Acquisition in Digitizing the HP141T	17
3.3.3 Design Considerations in Satisfying Display Requirements	21

3.3.4	Equipment for Debugging and Testing	25
3.4	System Design	27
3.4.1	Overview System Block Diagram	27
3.4.2	System Modularization	28
3.4.3	Digital Processing Subsystem	49
3.4.4	Graphical User Interface Subsystem	53
3.5	Experimental Test Procedures	54
3.5.1	Unit Tests for Subsystem Verification	55
3.5.2	Integration Tests for Overall Digitized HP141T System Verification	60
3.5.3	Acceptance Tests	60
4	Results and Discussion	61
4.1	HP141T Emulator Subsystem Test Results	61
4.2	Signal Conditioning Subsystem Results	61
4.3	Data Acquisition Subsystem Results	62
4.4	Digital Processing Subsystem Results	62
4.5	Graphical User Interface Results	62
4.6	Integrated System Performance	62
5	Conclusions	63
6	Recommendations	64
6.1	Hardware Recommendations	64
Bibliography		65
A	Appendix	67
A.1	Data Acquisition Subsystem Code	67
A.1.1	DAS Macros	67
A.1.2	Implementing Direct Memory Access for DAS	68
A.2	Digital Processing Subsystem Code	68
A.2.1	Using Piecewise Cubic Polynomial Interpolation to Smooth Display	68
A.3	Graphical User Interface Code	70
A.3.1	Pause Functionality Implemented on the Single-Board Computer	70
A.3.2	Enabling the Peak Hold Mode (PHM), Average Mode (AvM) and Raw Mode (RwM) Display Modes	70
A.3.3	System Documentation and Tutorial for Users	71
B	Appendix	73
B.1	Project Management Documentation	73
B.1.1	Gantt Chart	73
B.1.2	Bill of Materials	74
B.1.3	GitHub Log for Version Control	75
C	Appendix	76

C.1 AI Usage History	76
C.1.1 Prompts and Results for Clarification	76
C.1.2 Prompts for Editing	78
C.1.3 Prompts for Debugging	78
D Appendix	79
D.1 Graduate Attributes	79
D.1.1 GA Table	79

List of Figures

2.1	The Z-G Diagram by developed by Rohde & Schwarz with a Schmidt chart for measuring complex parameters [1].	6
2.2	The HP8568A spectrum analyzer which was the first spectrum analyzer from Hewlett-Packard Company to include a microprocessor in the 1970s.	7
2.3	Illustrating the size of early spectrum analyzer setups. Digitization and modernization of these devices was also focused on reducing their size in order to improve portability.	7
2.4	The Agilent 8560EC is shown with a LCD display. This was an improvement over the 8560E model which had a CRT display.	8
3.1	Methodology overview showing different stages in the iterative design process that was applied as a variation of the V-Model.	13
3.2	Heterodyne Spectrum Analyzer Block Diagram.	15
3.3	Vector analyzer block diagram showing digitization of the IF frequency.	16
3.4	HP141T CRT display section.	24
3.5	Overview system block diagram for the modernized HP141T which employs the HP8555A and HP8552B plug-in sections.	27
3.6	Calibration system diagram where the HP141T Emulator is used to configure the primary subsystems such as the SCS, DAS, etc.	28
3.7	Expansion of the overall system diagram showing the primary components and functions in each subsystem.	28
3.8	The HP141T Emulator subsystem diagram showing the primary components powered by the same ± 12 V DC dual rail supply.	30
3.9	Emulator circuits for vertical and horizontal outputs of the HP8552B IF section.	31
3.10	Simulated output at the STO pin of the XR2206 chip employed in emulating the vertical output of as a 3 V_{pp} sinusoidal wave.	31
3.11	Illustration of the duty cycle and rise time of sawtooth function corresponding to the horizontal output, V_x , of the HP8552B.	33
3.12	XR2206 sawtooth output level shifter for achieving the appropriate voltage range.	34
3.13	Simulated level shift circuit.	34
3.14	Showing simulated scan output of the emulator.	34
3.15	Simulation of the pen lift output using the XR2206 SYNC pulse output.	35
3.16	Pen lift output simulation with the XR2206 SYNC pulse in green and the pulse at the collector of the NPN transistor in red.	35
3.17	3D view of the HP141T Emulator subsystem's PCB.	36
3.18	HP141T Emulator power supply circuit with decoupling capacitors to reduce high frequency noise.	36

3.19 Schematic of the input connectors that enable the SCS to interface with the outputs from the HP8552B plug-in or HP141T Emulator subsystem.	37
3.20 Signal Conditioning Subsystem diagram.	37
3.21 The HSC pipeline uses an LM358 op-amp to process the horizontal output signal from the HP141T scaling and shifting it to match the sampling microcontroller's ADC's 0 V to 3.3 V input range.	38
3.22 Circuit schematic of the connectors with 0 V to 3.3 V outputs corresponding to the digitized values of the vertical, horizontal, and pen-lift outputs of the SCS.	38
3.23 HSC simulation output showing that the op-amp configuration is expected to maintain the shape of the sawtooth scan output.	38
3.24 An inverting amplifier configuration is used in the VSC pipeline to prepare the vertical output of the HP8552B for digital processing.	39
3.25 Simulation output for the case where V_{vin} assumes all values in the range between -0.8 V to 0 V.	39
3.26 Simulation shows significant clipping of the output at 0 V when V_{vin} has a peak-to-peak voltage of -0.32 V.	40
3.27 Simulation results indicate that the VSC circuit has a linear sensitivity of 70 μ V, compared to the linear sensitivity of 0.1 V in the vertical output of the HP8552B.	40
3.28 Showing reduced distortion and voltage clipping in the output for a vertical output frequency of 100 kHz.	41
3.29 At the maximum rated vertical output frequency of 300 kHz, the phase difference between the input and the output of the VSC is slightly more than π rad.	41
3.30 Schematic of the PLSC circuit for preparing the HP8552B pen-lift output for digital processing.	42
3.31 Simulation result corresponding to the maximum operation voltage of the PLSC circuit corresponding to 18.77 V.	42
3.32 The SCS is implemented on a single PCB with a single 9 V power source.	42
3.33 DAS subsystem block diagram showing the digitization of the analog signals using 2 16-bit ADCs operating in double-interleaved mode.	43
3.34 ADC1 and ADC2 convert the voltages in a period of 7 ADC clock cycles on the same channel.	44
3.35 Simulation results of a well-behaved sinusoidal signal that is sampled by the ADCs on the STM32H723ZG development board.	44
3.36 Simulation of a noisy signal sampled in double-interleaved mode by ADC1 and ADC2 on the STM32H723ZG.	45
3.37 Illustrating how the outputs of the HP141T can be sampled using input channels A and B on the Picoscope 2204A.	49
3.38 Flow diagram of the Digital Processing Subsystem which prepares time domain data for display in the frequency domain through the Graphical User Interface Subsystem.	50
3.39 Parts of the RPI 4B that are of particular interest for this project.	52
3.40 Illustrating the output voltage range associated with each subsystem. The operation performed by the subsystem is shown above the arrow preceding the block containing the output range of that subsystem.	52

B.1	Gantt Chart employed as a project management tool.	73
B.2	Part of the Bill of Materials for tracking component purchases.	74
B.3	Snippet of git log since the first commit.	75
C.1	AI was used for clarifying terms regarding the HP141T system before research was performed.	76
C.2	AI was used for clarifying terms regarding the HP141T system before research was performed.	77
C.3	AI was used to restructure phrases.	78

Abbreviations

ADC Analog-to-Digital Converter

AMPS Advanced Mobile Telephone System

API Application Programming Interface

ASIC Application Specific Integrate Circuit

AvM Average Mode

BLE Bluetooth Low Energy

BNC Bayonet Neill–Concelman

CPU Central Processing Unit

CRT Cathode Ray Tube

DAS Data Acquisition Subsystem

DC Direct Current

DFT Digital Fourier Transform

DMA Direct Memory Access

DPS Digital Processing Subsystem

DSI Display Serial Interface

DSP Digital Signal Processor

DVI Digital Video Interface

EDA Electronic Design Automation

ENOB Effective Number of Bits

EVSIS Emulator Voltage Scaling and Interpolation Schema

FFT Fast Fourier Transform

FIFO First In First Out

FPGA Field Programmable Gate Array

GPIO General Purpose Input/Output

GPU Graphical Processing Unit

GSPS Giga Samples Per Second

GUI Graphical User Interface

GUIS Graphical User Interface Subsystem

HAL Hardware Abstraction Layer

HDL Hardware Description Language

HDMI High Definition Multimedia Interface

HP Hewlett-Packard Company

HSC Horizontal Signal Conditioning

HVS Hardware Video Scaler

IF Intermediate Frequency

LCD Liquid Crystal Display

LED Light Emitting Diode

LO Local Oscillator

MCU Microcontroller Unit

MSPS Mega Samples Per Second

NVIC Nested Vectored Interrupt Controller

OLED Organic Light-Emitting Diode

OS Operating System

PCB Printed Circuit Board

PHM Peak Hold Mode

PLSC Pen-Lift Signal Conditioning

RAM Random Access Memory

RBW Resolution Bandwidth

RF Radio Frequency

RGB Red Blue Green

RPI Raspberry Pi

RTL Register Transfer Level

RTSA Radio-Time Spectrum Analayzer

RwM Raw Mode

SA Signal/Spectrum Analyzer

SBC Single-Board Computer

SCS Signal Conditioning Subsystem

SD Secure Digital

SDK Software Development Kit

SNR Signal-to-Noise Ratio

SPI Serial Peripheral Interface

SPS Samples Per Second

SVSIS Standard Voltage Scaling and Interpolation Schema

TFT Thin-Film Transistor

UI User Interface

USB Universal-Serial Bus

VGA Video Graphics Array

VSA Vector Spectrum Analyzer

VSC Vertical Signal Conditioning

Chapter 1

Introduction

1.1 Background

Designed and patented in the 1970s, Hewlett-Packard Company's (HP) high performance plug-in model 8552B and 8555A spectrum analyzers (SA), equipped with the 141T display, remain powerful tools for characterising signals in the frequency domain. The 8552B is particularly convenient for measuring spectra in a wide frequency range between 20 Hz to 40 GHz. Another advantage of these spectrum analysers is that a user can broaden frequency requirements by increasing the number of tuning sections. Additionally, the 141T features absolute calibration of amplitude as well as high resolution, sensitivity and a simple display output.

The shortcoming of the spectrum analyzer, however, is that it uses a cathode ray tube (CRT) display which is prone to degradation after extended periods of usage and is outdated compared to the display on most modern devices. In this project, a single board computer and a liquid crystal display (LCD) touch screen is interfaced with the 141T display unit, thereby, replacing the outdated CRT technology. This allows users to continue to exploit the advantageous capabilities of the spectrum analyzer, such as the wide frequency bandwidth, despite damage to the CRT display. In addition, interfacing a single board computer with the 141T offers improved software-based features for performing frequency analysis.

Specifically, this project aims to develop a new display for the high resolution 8552B intermediate frequency (IF) section equipped with the 8555A spectrum analyzer radio frequency (RF) section which can make frequency domain measurements from 10 MHz to 18 GHz. The broad scanning frequency bandwidth of this model makes it suitable for frequency domain analyses in engineering applications such as mechanical vibrations and EMC field strength analysis with a calibrated antenna [2].

The CRT display subsystem consists of a post-accelerator storage tube with a 9 kV accelerating potential and aluminized P31 phosphor for producing high trace brightness. When calibrated, the CRT screen can display frequency bandwidths of up to 2 GHz wide. To display the full frequency range with a maximum of 18 GHz, the CRT can be calibrated in 10 frequency bands using internal mixing. One of advantage of the 141T over other displays that were manufactured during that time is that more detail can be observed in the spectrum by progressively narrowing frequency width from 100 Hz/division to 2 kHz/division. Overall, the 141T consists of a CRT graticule which can plot the frequency domain representation of a signal on a 2D plane with 8 x 10 divisions.

For this project, the 141T is powered by a 220 V single-phase source at 60 Hz, requiring less than

225 W even when plug-ins are connected. To achieve the 9 kV accelerating potential for deflecting electron beams in producing the **CRT** display, the device uses a step-up transformer and transistorized oscillator. The main disadvantage of having to increase the accelerating potential in a **CRT** display system is that the performance of electronic voltage regulation components such as capacitors, diodes and resistors can degrade over time.

Another challenge of using a phosphor **CRT** display is the effect of persistence on the saccadic information transfer which can lead to bias in experimental results [3]. This effect of persistence on experimental results is of particular interest to frequency domain analysis since displayed signals include noise from the environment which can make it difficult to extract accurate frequency information from plots. For the Model 141T, the persistence varies from the natural persistence of P31 phosphor (0.1 s) to a maximum of 15 s when the device is operating in the maximum writing rate mode. Therefore, phosphor persistence in the **CRT** display can significantly affect the amount of time to acquire measurements as well as the precision of the data extracted from the display.

1.2 Objectives

The aim of this project is to design a new display with full functionality and computer-aided signal processing features such as signal normalization. The digital display has to be compatible with the voltage outputs that enable analog signals to be plotted by the 141T. The aim of digitizing the signals is to interface measures from the spectrum analyzer with a computer that can perform tasks and store data accordingly. Therefore, a survey of the 8555A **RF** section and 8552B **IF** section outputs and available single board computer and touch screen options must be conducted. Furthermore, the project aims to investigate basic XYZ replicas, performing digital signal processing algorithms, how to correctly display signal data on annotated axes depending on available instrument settings.

This report aims to provide:

- Characterization of the HP141 display inputs from the 8555A **RF** section and 8552B **IF** section
- Available options for single-board computer and touchscreens options and the most suitable selection for interfacing with the two spectrum analyzer sections
- A design and simulation of interface between the single-board computer which includes analog converter for digitizing outputs from the **RF** and **IF** sections
- Algorithms for processing the digital signals and performing operations on displayed spectra
- Results on the construction, unit tests and integration tests of the improved system for spectral analysis

1.3 Project Requirements

Before detailing system requirements, user requirements were used to scope the project in terms of objectives that are not related to functions and performance. In designing the upgraded or ‘new’ **SA**, selection of hardware components was conducted with the aim of formulating specifications that

successfully fulfill user requirements. Table 1.1 summarises the user requirements and gives a short description of the objective.

Table 1.1: User requirements.

ID	Requirement Description
UR01	Display of the new SA must behave like the display of newer generations of SAs, such as the FieldFox. That is, the new SA must achieve more or less the same number data points as the FieldFox (801 points).
UR02	The SA must have the following display modes: (a) Peak hold mode (PHM) which displays the largest value seen and updated at each scan. (b) Average mode (AvM) in which each frequency bin's average is updated at each scan. (c) Raw mode (RwM) where the latest value is displayed until the next scan and overwriting each value during a scan event.
UR03	SA unit must have a suitable vertical resolution based on a 10 dB/division in the logarithmic scale.
UR04	Linear display mode must have low priority.
UR05	SA display subsystem must have setting markers, similar to the FieldFox analyzer.
UR06	Design must be capable of storing and recalling traces.
UR07	Users must be able to change modes by touching the screen and must be able to enter data using a keyboard.
UR08	All software must load on power up.
UR09	SA unit must use a single wall wart power source for the display subsystem.
UR10	Project must develop an HP141T display emulator of vertical amplitude, horizontal sawtooth and pen lift state.

Table 1.2 details system requirements in terms of functions and performance. These requirements were developed after a review of the scope through the formalization of the user requirements in table 1.1 above.

Table 1.2: System requirements.

ID	Requirement Description
SR01	The system must digitize analog outputs from an HP141T 8555A model which performs frequency domain measurements between 20 Hz and 18 GHz.
SR02	Digitized outputs must be interfaced with a single board computer for performing signal processing tasks.
SR03	The system must include a signal conditioning box for debugging purposes and replicating the outputs of the spectrum analyzer during testing.
SR04	The system must be simulated using software.
SR05	The display must include new annotations that take instrument and operator manual inputs into account.
SR06	The system must include appropriate documentation such as tutorials and operational instructions when using the signal processor and screen.

In addition to the above mentioned system requirements, the project considers the basic configuration parameters that signal analyzers typically provide such as:

- Setting the minimum and maximum frequencies to be displayed based on a given center frequency

- Setting the reference amplitude for frequency responses and a span that is suitable for the spectrum analyzer
- Setting the frequency resolution according to the passband of the **IF** filter
- Setting the sweep time required to record the full frequency spectrum that is of interest

1.4 Scope & Limitations

The focus of this report is in the design and implementation of a digitized display for the HP141T that interfaces with a single board computer for storing and manipulating signal data from the oscilloscope. The scope is limited to selection of electronics that are suitable for converting the analog signals from the HP141T that are responsible for displaying signals. The scope only includes a survey of the HP141T circuits and outputs that directly affect how a spectrum is generated with respect to time domain and frequency domain analyses. The paper is not concerned with changing or improving the operational design of the device with respect to its power, amplification and filtering circuitry.

1.5 Report Outline

Chapter 2 initiates the report by establishing the general history and theoretical framework for the design and applications of spectrum analyzers in engineering. The same chapter details the previous techniques for converting the output of a **SA** to digital values that can be manipulated by a processor. Finally, designs of displays are explored in literature to establish an approach to representing the processor output on a **LCD** screen.

Chapter 2

Literature Review

The aim of this chapter is to conceptualize the operation of spectrum analyzers and establish a theoretical foundation for the frequency analysis techniques applied to produce the correct output. This conceptualization is then integrated with a broader review of digitizing and modernizing the display of spectrum analyzers.

In circumventing design limitations of spectrum analyzer displays, it is prudent to survey the most suitable hardware components. This is particularly true for the case where electronic components are required to perform in a broad frequency bandwidth. For example, for high frequency signals, the Nyquist theorem indicates that the ADC is required to have a sample at a frequency that is more than double the frequency of the output signal. Furthermore, the challenge of presenting signals in the frequency domain using electronics exists due to the fact the input signal to the ADC holds information about frequency in the time domain. Therefore, the investigation of literature that is presented in this chapter aims to provide a motivation for the design decisions taken in digitizing and modernizing the HP141T display.

The chapter begins with an evaluation of the frequency domain analysis theory that is applied in the operation of signal analyzers. Then, the different principles that distinguish different types of analyzers are explored to form the basis understanding the expected behaviour of a spectrum analyzer with specific settings. Following descriptions of the operation of spectrum analyzers from literature, the chapter includes a review of the investigation into different techniques for digitizing analyzer displays. This also includes a review of the different electronic components and techniques for digitizing frequency information in order to survey available hardware options that can be selected for a cost effective implementation. Finally, a broad discussion is included on different types of displays for analyzers in literature and a critique of the literature is provided to outline the purpose of the proposed design.

2.1 Mitigating Spectrum Analyzer Obsolescence: A Historical Perspective

2.1.1 Introduction to Spectrum Analysis

A spectrum analyzers [SAs](#) is a critical instrument for investigating properties of physical phenomena that can be interpreted through power and frequency characteristics of analog signals derived from voltage measurements. This project differentiates between spectrum analyzers and oscilloscopes such that spectrum analyzers are instruments that display waveforms in the frequency domain and oscilloscopes

2.1. Mitigating Spectrum Analyzer Obsolescence: A Historical Perspective

as instruments that operate in the time domain. However, the paper recognizes their significance in signal processing applications, as well as some of the shared similarities in their subsystems.

As such, the following section is a review of the history and applications of spectrum analyzers and oscilloscopes to establish the context of the project. By assimilating literature on the applications of these instruments in engineering, the section also aims to formulate design guidelines from previous works that can improve the operation of the digitized HP141T, thereby satisfying user and functional requirements.

2.1.2 Early Developments in Waveform Analysis

The evolution of [SAs](#) is closely linked to that of the oscilloscope. In a review of the history and technology of oscilloscopes, Pereira attributed the invention of the electromagnetic oscillograph to the French physicist, André-Eugene Blondel [4]. Oscillographs were devices that used a pen attached to a moving coil to trace an ink record on a rotating paper chart [4]. According to Herres, the motivation for the invention of oscillographs was to extract waveforms from acoustic and electrical phenomena, however, the devices had a severely limited frequency response and bandwidth because of the inertia of the pen and ink recording equipment [5]. Ultimately, the operation of these devices was restricted by the working principle based on mechanical devices which limited the bandwidth in the range of 10-19 Hz [4].

Following the invention of the oscillograph, available waveform analysis tools included contact diode and vacuum tubes as the only form of signal detectors [6]. Technologies such as the slotted line and phase bridges became the standard instruments for measuring amplitude and frequency but were often slow and tedious [1]. In 1933, Rohde & Schwarz developed the improved Z-G system which was the first to directly indicate complex parameters on a Schmidt chart as shown in figure 2.1 [1].



Figure 2.1: The Z-G Diagram by developed by Rohde & Schwarz with a Schmidt chart for measuring complex parameters [1].

2.1.3 Advancements in Spectrum Analysis Technology

The first commercially available real-time signal analyzers ([RTSAs](#)) were introduced in the 1960s by Federal Scientific to process data up to 20 kHz through a single filter that could change modes in milliseconds [7]. These devices were designed for analyzing mechanical faults and failures in rotating machinery and for investigating vibratory motions of components, systems and structures [7]. In contrast to the the [SAs](#) introduced in the 1960s for characterizing rotary machinery, this paper is focuses on [SAs](#) that were designed for displaying wide-band signals up to the K-band of microwave signals.

Rapid progress in semiconductor technology and microwave elements laid a foundation for the developed first waveform instrumentation devices with 1 GHz bandwidth [6]. With the advent of wireless network

2.1. Mitigating Spectrum Analyzer Obsolescence: A Historical Perspective

technologies, **SA** developments became focused on high frequency instrumentation which had to meet testing requirements of first generation technologies such as the **Advanced Mobile Telephone System (AMPS)** operated at frequencies at 800 MHz to 900 MHz. Vector network analyzers were introduced as an extension of spectrum analyzers capable of displaying amplitude and phase relative to a reference signal [8].

2.1.4 Early Modifications to Spectrum Analyzers (1960s - 1980s)



Figure 2.2: The HP8568A spectrum analyzer which was the first spectrum analyzer from Hewlett-Packard Company to include a microprocessor in the 1970s.

a plug-in polar **Cathode Ray Tube (CRT)** display to show amplitude and phase [10]. The aim of the display was so that the device could be used with Smith chart overlay, or reflection coefficient measurements [10].

In 1982, D'Addario proposed the implementation of Hewlett-Packard Model 8410A network analyzer as a reflectometer in a system which included an Apple II Plus computer for performing computations of the short-open-load method of calibrating and correcting errors of signals in the S-band of the microwave region of the electromagnetic spectrum [11]. D'Addario stated that the advantage of implementing a computer-based system was in the software substituted a short delay line for open circuit calibration [11]. In addition, the software was mostly in Applesoft Basic that featured assembly language subroutines for performing complex arithmetic and controlling the interface [11]. Similarly, the proposed design in this project aims to exploit the advantages of computerization by introducing functions and libraries for performing digital signal processing tasks and controlling the **Graphical User Interface (GUI)**.

Another modification to the HP8410 was performed by NASA's Terry and Kunath in 1990, where the **SA** was used as an automated far-field antenna range receiver [12]. The

Motivated by applications in network analysis and microwave engineering, Hewlett-Packard developed the first fully calibrated spectrum analyzer that capable of sweeping broad frequency ranges [9]. After the introduction of easily-operated wideband network analyzer equipment in 1967, Hewlett-Packard Co. released a network analyzer with a range of 100 kHz to 110 MHz. In particular, the company released the Model 8407A which could resolve amplitude of up to 0.05 dB on



Figure 2.3: Illustrating the size of early spectrum analyzer setups. Digitization and modernization of these devices was also focused on reducing their size in order to improve portability.

2.1. Mitigating Spectrum Analyzer Obsolescence: A Historical Perspective

system included external mixers capable of harmonic mixing up to 18 GHz, Analog-to-Digital Converter (ADC)s, and interfaced with an external computer [12]. Terry and Kunath noted that the phase and amplitude signals could be converted to digital signals by using either ADCs or by using a digital voltmeter [12]. The authors used a HP308 personal computer with two processors that operated in the DOS 3.3 and HP Basic 5.0 environments, respectively [12]. The computer was employed as the system controller with programs for creating and editing control files for swept frequency, viewing radiated power level data, and running tests with parameters saved in files to be executed or edited at a later time [12]. Consequently, the designer needs to consider file formats, file sizes and file transmission in the process of digitizing a SA. This is particularly true when the system processes are performed across different devices and programs which may support different file formats and file sizes.

2.1.5 Commercial Digitization of Spectrum Analyzers

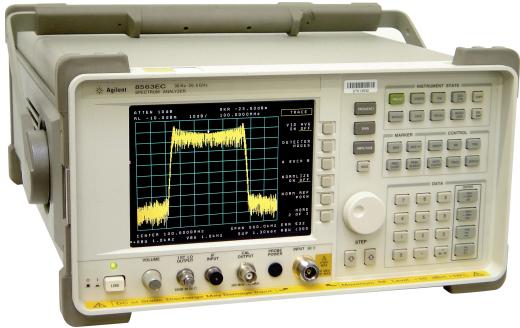


Figure 2.4: The Agilent 8560EC is shown with a LCD display. This was an improvement over the 8560E model which had a CRT display.

over previous versions when the HP8568A automatic SA, shown in figure 2.2, was introduced with a microprocessor controlling its operation.

In the 2000s, Agilent Technologies modernized the front and rear panels of HP8560E Model spectrum analyzers, by introducing LCD color display interfaced with a LCD driver board and VGA port located that did not require a user interface [14]. Further additions included ADC circuitry integrated into the controller board [14]. These improvements also translated into better portability of the system as shown in 2.4.

2.1.6 Obsolescence Mitigation and Modernization Strategies

Overtime as prices of new spectrum analyzers increased rapidly, researchers opted to use older SAs instead of replacing them. Research into strategies for mitigating risks related to the obsolescence of older SA models was conducted by Hoppin in 2002, where particular focus was allocated to HP8566/68 SAs in automated test equipment [15]. Hoppin stated that dependence on obsolete and unsupported equipment increased the risk of system and that as SAs aged, the time required for repair and the cost of replacement increases rapidly [15]. As a result of the inevitable obsolescence, Hopping proposed three strategies for dealing with the aged devices, namely, the *stockpile*; the *redesign*; and

HP85XX spectrum analyzers replaced the HP84XX analyzer series as the first SAs with synthesized frequency sweeps and an ADC converter for sampling a video signal and displaying it on a CRT screen. The HP85XX series expanded the capabilities of spectrum analyzers, however, earlier versions of these devices were large as shown in figure 2.3, where a technician is shown alongside a HP8566B SA at the heart of two EM1 receivers [13]. Significant improvements to this series of SAs were realised

2.1. Mitigating Spectrum Analyzer Obsolescence: A Historical Perspective

migration strategies. The stockpile strategies requires industry organizations to identify sources of spare replacement components and instruments, and offer repair and calibration services [15].

This project follows a strategy similar to the redesign strategy proposed by Hoppin in which automated test equipment program managers are required to assess the feasibility of introducing replacement instrument specifications, feature and capability requirements [15]. Hoppin noted that one of the risks of applying the redesign strategy is that software projects are notoriously hard to manage to scope, budget and schedule [15]. Additionally, since the devices have aged significantly, the languages, tools and methods originally used may be lost, requiring designers to apply reverse engineering techniques [15]. To mitigate the costs from applying the redesign process, Hoppin described three main components that need to be considered, namely, test code creation, test instrumentation, purchase and integration, and revalidation measurements and results [15].

In a similar research published in 2010, Wolle discussed the various aspects related to replacement of obsolete instruments to overcome the challenges that arise when newer versions are not compatible with previous systems [16]. In the paper, Wolle postulated that an emulation strategy can help designers to incorporate new test instruments to migrate from obsolete instruments [16]. In addition, Wolle noted that when assessing version migration strategies, modernizing test equipment typically has much higher costs but leads to greater reliability and faster tests [16]. This was first noted by Hoppin who detailed two risks of implementing a migration strategy as low compatibility and disagreement in measurement results due to differences in architectural and measurement methods between legacy and newer-generation platforms [15].

In a 2014 paper by Iglesias et al., the high cost of spectrum analyzers is also noted as a primary motivation for further developing spectrum analyzer designs [17]. Iglesias et al. also indicated that because **SAs** employ non-invasive methods to detect machine failures in real-time, the devices need to be continually upgraded to reduce failures by improving the efficiency of industrial monitoring systems [17]. Following the motivation for improving **SAs**, Iglesias et al. postulated that modernization of **SAs** was primarily enabled by the fast evolution of **ADCs** and **Digital Signal Processor (DSP)**s which offer the opportunity to implement multiple methods for spectrum estimation such as Barlett and Welch's methods for averaging periodograms [17]. The authors listed devices such as **Field Programmable Gate Array (FPGA)**s, **Application Specific Integrate Circuit (ASIC)**s, general-purpose **CPUs**, and **Graphical Processing Unit (GPU)**s as feasible candidates for digitizing **SAs** [17], however, **ASICs** were listed as the best implementation with respect to area, power, and speed [17].

2.1.7 Conclusion

In conclusion, the evolution of spectrum analyzers is characterized by advancements in metrological accuracy in waveform analysis, a migration from analog to digital processing, as well as improvements in functionality and hardware of the user interface. From early mechanical oscilloscopes to modern digital spectrum analyzers, each technological advancement has been motivated by the need for higher frequency capabilities and improved usability. Technological advancements have culminated in the introduction of computerized control using microprocessors and **ADC**-based systems that support more efficient signal analysis in the digital time domain.

A primary conclusion that can be drawn from the historical review of **SAs** is that modernization

2.2. A Preview of the Operation of Spectrum Analyzers to Establish the Context for Digitization and Modernization

strategies for obsolete instruments must balance cost, functionality, and compatibility with newer technologies. **SAs** and network analyzers like the HP8410A and HP8568A introduced microcontrollers to automate functions. More recent trends that were introduced with Agilent Technologies' **SAs** emphasized **GUIs**, software-based processing, and integration with digital communication protocols.

In this project, the digitization of the HP141T spectrum analyzer builds on the primary ideas presented in history on the modernization of spectrum analyzers. Similar to implementations in history, the design proposed in this paper leverages modern single-board computing, digital displays, and software-based signal processing techniques for implementing the redesign strategy for circumventing the obsolescence of the **SA**. Furthermore, this project aims to retain the core measurement functionalities of the original device while introducing intuitive graphical interfaces and improved data handling capabilities. The following sections discuss similar works and the challenges that researchers overcame in digitizing and modernizing spectrum analyzers.

2.2 A Preview of the Operation of Spectrum Analyzers to Establish the Context for Digitization and Modernization

Formulating an understanding of the internal operation of the HP141T system is critical for extending functionality through digitization and modernization. In this regard, this project establishes the context of the term **digitization** as the development of a system which converts the analog signals from the **SA** into digital values that are processed on a computer. The context for the term **digitization** and its usage were explored through literature as described in this section. Key related works were collated and focus is placed on detailing the internal path of an analog signal through the major components of **SAs** in general.

The term *modernization* is also clarified through collated literature. In particular, this section details features of modern **SAs** and their user interfaces which largely depend on the employed display type. By redesigning the display of the HP141T, the project aims to improve the overall performance of the system with respect to power consumption, ease-of-use, portability, and signal processing features. Changing the display from **CRT** to **LCD** can be considered as a partial fulfilment of modernization, while ensuring that analog outputs from the HP141T can be stored and retrieved for processing at a later time is forms part of the digitization of the system.

To establish a context for these terms, this section begins by differentiation between the different types of **SAs** while focusing on the key operational principles that govern the behaviour of the HP141T. Then, the section details the purpose of primary components in spectrum analyzers to formulate a steps for digitizing the system from literature. The section concludes with findings on features of modern spectrum analyzers that differentiate them from older models, and challenges that related works overcame during the modernization process.

2.2.1 Classifications of Spectrum Analyzers

2.2.2 Primary Components of Spectrum Analyzers

2.2.3 Features of Modern Spectrum Analyzer Displays

2.3 Digitizing Spectrum Analyzer Outputs

2.3.1 Output Voltage Regulation and Preparation for Frequency Analysis

2.3.2 Transforming Spectrum Analyzer Output Signals to Digital Frequency Domain

2.3.3 Interfacing Computers with Spectrum Analyzers

Chapter 3

Methodology

3.1 Methodology Outline

This chapter details the design process and approach employed in achieving the aim of the project which is to digitize and modernize the HP141T display by replacing the [CRT](#) monitor with a [LCD](#) touchscreen display that offers different functions and modes of operation. Design decision are also documented here, showing the considerations that were made based on the operation and outputs of the HP141T spectrum analyzer and ensuring that the newly integrated display is compatible with the device's hardware. For example, the design required selection of the digital hardware for processing the analog voltage signals from the [SA](#). The selection of the digital processor made from single-board computers such as the Raspberry Pi 4 Model B, microcontrollers like the STM32F4 boards, [FPGA](#) such as the Artix-7 from Xilinx, or a heterogeneous digital processor which consists of a combination of these options.

Other considerations were made regarding the electronic circuits for converting the auxiliary output voltages from the HP141T to the appropriate voltage level for the operation of [ADC](#). This chapter describes how together, the [ADC](#) and digital processor form a crucial part of the system. Additionally, the chapter details the software development kit ([SDK](#)) and associated coding language that was used. The selection of the development framework depended on the choice of processor and digital processing algorithms that were required to fulfil the project requirements. For example, assuming that a [FPGA](#) is the chosen digital processor, the [SDK](#) would include tools such as the AMD Vivado and electronic design automation ([EDA](#)) software and the choice of coding language between Verilog, VHDL, and SystemVerilog would depend on how comfortable the developer is in representing digital processing algorithms, such the [DFT](#), using the chosen language.

Overall, this chapter documents an overview of the design methodology and different phases in the design process. The design process followed a variation of V-Model in which a series of iterative phases was implemented with a process checking mechanism. The chapter begins by highlighting the design stages and modularization of the system. Thereafter, the chapter includes an assessment of the project requirements detailed in introductory section. Finally, the chapter describes the use of findings from the review of requirements in informing the design decisions and specifications.

3.2 Phases in the Design Process

The design process was decomposed into four iterative stages as illustrated in Figure ?? below. The first stage documented the digitization requirements and listed examples of modern **SA**s in order to define the requirements for modernization. As shown in the methodology overview diagram, the first stage also included thorough investigation into methods that have been implemented in literature for upgrading the functions and performance of **SA**s. A theoretical framework for relevant information about signal processing was also formulated based on the literature to ensure that the display modes and functions were consistent with the mathematically derived expected outputs in the system.

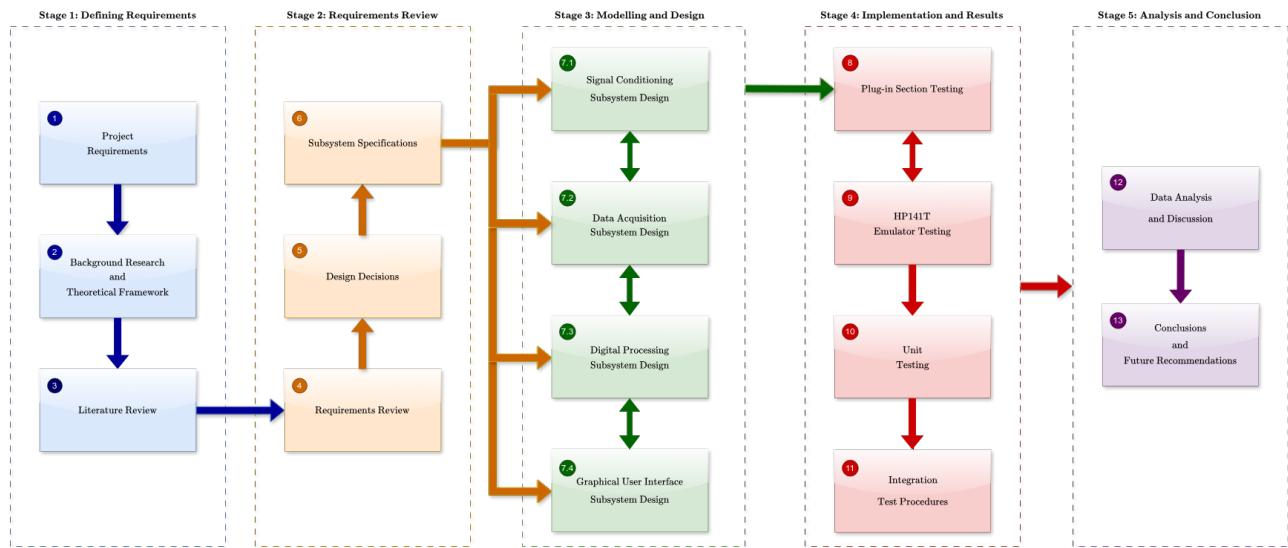


Figure 3.1: Methodology overview showing different stages in the iterative design process that was applied as a variation of the V-Model.

The second stage in the design process involved a review of the user requirements specifications detailed in chapter 1. The aim of this step in the methodology was to clarify the desired functions of the upgraded spectrum analyzer and to inform the design decisions with respect to the digital hardware and software development for digital signal processing. Additionally, the requirements review was employed in modularizing the design into four subsystems including, the Analog-to-Digital Conversion, Digital Processing, Screen and User Interface subsystems.

Stage 3 of the design process dealt with the design specifications of each of these subsystems by further decomposing each module into smaller hardware and software components. Each function is tested against a requirement in Stage 4 where each design specification was verified and the operation of the integrated system was tested to ensure that the interfaces between the subsystems was configured correctly.

In general, implementation of the design process adhered to the following design steps detailed in the project brief and are associated with the user requirement specifications:

1. Surveying the HP141T display and other outputs.
2. Surveying single-board computers and touchscreen options.

3. Phase One: establish a basic XYZ replica or HP141T emulator.
4. Phase Two: implement averaging and peak hold options.
5. Phase Three: annotate display axes.
6. Phase Four: determine the appropriate instrument settings.
7. Phase Five: design new annotations for display taking instrument or operator manual inputs into account.
8. Phase Six: include more tutorial and operational instructions using.

In finalizing the design process of the upgraded SA, results from acceptance tests were assimilated and a conclusion was drawn. Then, based on the outcomes of the project, future recommendations were made for future iterations of the upgrade SA system. Overall, the stages of the methodology included:

- Stage 1: Defining Requirements
- Stage 2: Requirements Review and System Overview
- Stage 3: Modelling and Design
- Stage 4: Implementation and Results
- Stage 5: Analysis and Conclusion

The first four stages of the design process were performed iteratively as a variation of the V-Model in which risk analysis was performed at each step, similar to the Spiral Model for design processes. Each iteration aimed at producing a new version of the upgraded HP141T display and a single conclusion was made when a satisfactory prototype was established.

3.3 Requirements Review

This section aims to clarify the scope of the project and produce a technical formalization of the user requirements specifications. The section begins by giving context to the term modernization by investigating the properties of modern SAs. Then, the section breaks down the user requirements into categories relating data acquisition, processing and display. Finally, the section concludes with a review of the requirement for developing a HP141T emulator.

3.3.1 Comparing the HP141T System to Modern Spectrum Analyzers

The first user requirement deals with the modernization of the HP141T system. The aim of this section is to review this requirement and to give clarification on the context of the use of ‘modernization’ in this project. To achieve this objective, the section begins with a description of the HP141T system and the functions that differ from ‘modern’ spectrum analyzers, such as the range of hand-held SAs by FieldFox, and the N9000B CXA SA manufactured by Keysight®. The section concludes with a summary table comparing the features of the HP141T system and modern spectrum analyzers.

The HP141T system is equipped with three plug-ins, namely the 8555A **RF** section which operates in the microwave region of the electromagnetic spectrum, the 8552B **IF** section and the Model 141T display which includes the **CRT** monitor. The three plug-ins operate in unison to electronically scan signals in the time-domain and provide a visual representation of the input signal's amplitude in the frequency domain on the **CRT** monitor [18]. Amplitude can be represented on a logarithmic (dB m) or linear scale (mV).

The 8555A and 8552B plug-ins apply principles of heterodyning spectroscopy, allows high frequency input signals with frequencies in the K-band of the microwave portion of the electromagnetic spectrum (i.e. 18 MHz) [19]. As noted from literature in the previous chapter, spectrum analyzers which rely on the **FFT** by sampling the continuous input signal and performing the Fourier transform on N total samples typically deal with signals that have lower frequencies. This is because **SAs** that employ the super-heterodyne principle like the HP141T determine the spectrum directly by analysis in the frequency domain and not from the time-dependent characteristics of the input signal. This is achieved by using a super-heterodyne receiver comprised of a mixer and tunable local oscillator (**LO**) which convert the input signal to an intermediate frequency as illustrated in figure ?? [20].

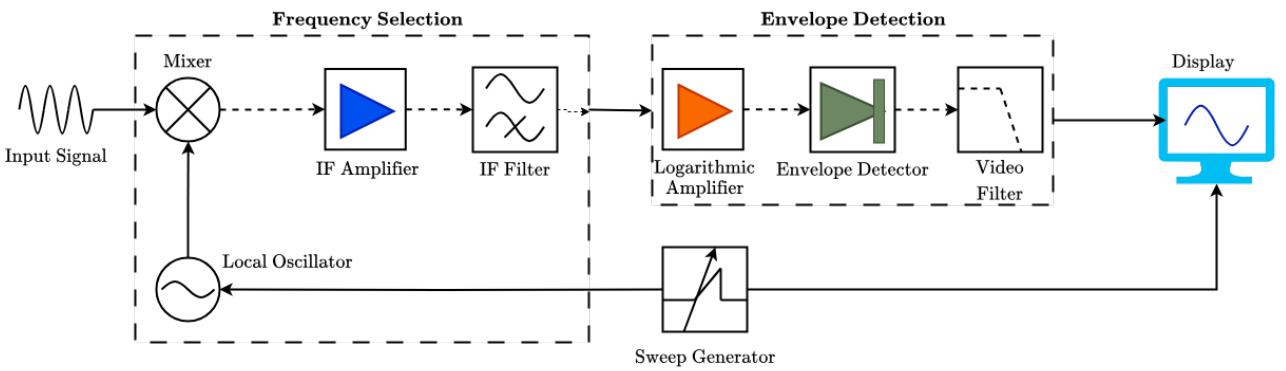


Figure 3.2: Heterodyne Spectrum Analyzer Block Diagram.

As noted in the literature review, most modern **SAs** are real-time spectrum analyzers **RTSA** which are a special case of vector spectrum analyzers (**VSA**) that use the super-heterodyne principle but digitize the input signal at the intermediate frequency, as shown in Figure ??, using a bandpass filter which can behave like a pre-select filter to limit distortions that arise from the mixer and shows the result in real-time. The advantage of digitizing the output of the **IF** section is that it enables large a input range of up to 50 MHz which can be analyzed in the time and frequency domain using digital signal processing techniques [21]. Much like **FFT** analyzers, vector analyzers are limited by the minimum and maximum sampling rate at which the **ADC** can operate. Additionally, exceedingly high sampling rates can lead to aliasing which causes frequencies outside of the bandwidth to fold into the frequency band [21].

Examples of modern real-time analyzers include:

- Tektronix RSA2208A which can scan input signals with frequencies of up to 8 GHz.
- Rohde & Schwarz FSP13 which operates within a 9 GHz to 30 GHz **RBW**.

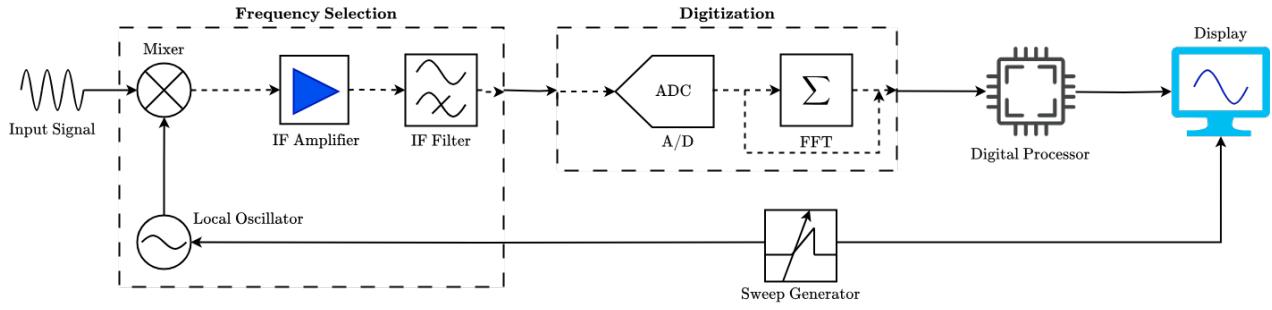


Figure 3.3: Vector analyzer block diagram showing digitization of the IF frequency.

- Agilent E4445A which offer frequency analysis up to 13.2 GHz.
- FieldFox RTSA which covers 5 kHz up to 50 GHz.

In modern **RTSAs**, **IF** analog output is transferred to the **ADC** through an anti-aliasing filter before being stored in memory [22]. This is in contrast to older heterodyne **SAs** such as the HP141T which takes the **IF** output through a video filter is essentially an averaging low-pass filter which smooths the **IF** output and reduces the effects of internal noise before the signal is channelled to the display [20]. HP141T video filters may select 10 Hz, 100 Hz, 10 kHz or OFF section of the low-pass filter for the detected video.

The output of the the HP141T video filter is transmitted to the vertical deflection of the **CRT** display [2]. For the horizontal deflection of the electron beam in the **CRT** display, a sawtooth signal from the sweep generator governs the horizontal scaling of the output frequency as electrons impinge on the phosphorescent screen. Modern **SAs** retrieve data from memory through high speed data lines to display information about the digital signal from the **ADC**. Digital display technologies include:

- **LCD** - most commonly used digital display which represents signal data with a sharp image and has low power consumption
- **OLED** - typically used in high performance **SAs** for displaying color coded spectrograms
- **TFT** - offer good visual quality and response time for sharper imagery when analyzing high frequency signals.

Displays in modern **SA** architectures offer more options for displaying frequency domain data. This is enabled by the processor, which is typically hosted on microcontroller board with high clock speed for performing digital signal processing algorithms. For example, **RTSAs** display the spectrogram as a color coded function of the amplitude of the signal [23]. In addition, digital displays generally consume less physical space compared to **CRT** displays with the same screen dimensions. This has allowed spectrum analyzers to become smaller as they entire system can fit into smaller packaging. For example, the FieldFox is an hand-held **RTSA** which despite its size, offers improved performance over the HP141T.

The comparison table in 3.1 below summarizes the differences between the HP141T analyzer (equipped with the 8555A and 8552B plug-ins) and modern **RTSAs**.

Table 3.1 shows that **RTSAs** also have connectivity abilities. For example, data can be transferred

Feature	HP141T	Real-Time Spectrum Analyzer
Operation	Heterodyne	FFT-based real-time processing
Frequency Range	10 GHz - 40 GHz	DC to 54 GHz
Scan Speed	0.1 ms/div - 10 s/div	Real-time GHz bandwidth capture with no losses
Dynamic Range	High but limited by analog filters	Very high due to digital signal processing
Resolution Bandwidth	Adjustable limited by analog filters	As low as 1 Hz
Real-time Capture	No. Displays active signals and struggles with pulsed signals	Yes. Captures transient and pulsed signals with ease
Signal Processing	Analog signal processing with IF filters and logarithmic amplifiers	Digital signal processing from ADC-sampled signal
FFT Capability	-	Built-in FFT
Display Type	CRT	LCD/OLED/TFT
Trace Storage	No memory.	Digital data storage
Connectivity	None	USB, Wi-Fi, Bluetooth, BLE
Size & Portability	Large and mountable on a rack	Compact and portable
Settings & Configuration	Manual tuning and calibration	Automated measurements, presets, user-friendly

Table 3.1: Comparing the features of the HP141T system to the modern RTSAs.

serially from on-board memory to an external device through the USB transfer protocol or through the BLE transfer protocol.

Features shown in the comparison formulated the modernization criteria to give context to the user requirement specification. Following sections expand on the features which can provide the functionality for associated requirements.

3.3.2 Data Acquisition in Digitizing the HP141T

As seen in the previous section, a primary criteria for modernizing HP141T is the translation of continuous analog input signals from the time domain to a digital signal that can be assessed using a digital processor such as a single-board computer, FPGA, or microcontroller. Selecting the point at which the conversion occurs in the pipeline of the HP141T super-heterodyne system plays a crucial role in the performance of the modernized system. The following project aimed to fulfil the signal conversion criteria by sampling the input signal at the intermediate frequency in a similar manner to the technique implemented by a modern RTSA and vector analyzers. In addition, user requirement UR01 states that the system needs to sample 801 points which is a standard for modern signal analyzers.

The following section reviews the user requirements to inform decisions about the data acquisition subsystem of the digitized HP141T subsystem. One of the most crucial parts in the design of the data acquisition subsystem is the selection of the ADC which samples signals and transmits them to the digital processor to store or manipulate signals in real-time. The section describes the limitations of this central part of the system and proposes solutions that can be implemented to fulfil the user requirements.

ADC Constraints

Table ?? shows constraints that influence the selection of the ADC and a solution for circumventing the limitation. For example, the operating voltage range of the ADC is a constraint which can be circumvented by including a signal conditioning circuit.

Constraint	Problem	Solution
Operating Voltage	Limited input voltage range of 0 V to 3.3 V or 5.0 V. The HP141T produces incompatible auxiliary output voltages	Signal conditioning circuit for scaling and shifting signals for compatibility with ADC operating voltages
Sample Rate	Required minimum of 801 points	Select an ADC with a high sampling rate (at 1 MSPS)
Resolution	Higher resolution is required for finer amplitude accuracy	Use an ADC with a minimum resolution of 16-bits
Number of Channels	The HP141T has 3 auxiliary outputs that are related to the frequency vs amplitude plot	Use an ADC with at least 3 channels
Latency	High latency can introduce delays in rendering signal data in real-time	Choose an ADC with SPI for low latency
SNR	ADC with a low SNR are more susceptible to noise and fluctuations in the amplitude	Use an ADC with a high SNR

Table 3.2: ADC selection constraints and solutions.

Conditioning HP141T Output Voltages to Match the ADC Operating Voltages

To satisfy the modernization requirement, at least three ADC channels are required to sample the three auxiliary analog output signals from the HP141T system that are shown in Table ?? below.

Name	Range	Description
Pen-Lift	0 V to 14 V	Triggers the CRT display. The pen-lift output is 0 V during scanning
Vertical Output	0 V to -0.8 V	Detected video output proportional to the vertical deflection on the CRT, corresponding to the amplitude of the signal in the frequency domain. Voltage range is scaled for full vertical deflection of the electron beam in 8 divisions of the CRT display
Horizontal Output	-5 V to 5 V	Sawtooth signal for representing scaling frequency across 10 divisions of the horizontal deflection on the CRT display. The HP141T uses this output to control the trace of the spectrum across the screen.

Table 3.3: Auxiliary voltages of the HP141T that must be sampled to digitize the system.

Since the operating voltage of most ADCs is in the range between 0 V and 3.3 V or ± 5 V, the auxiliary outputs listed in table ?? need to be scaled and shift to be within the operating voltage range. To scale the signal, an op-amp based circuit with appropriate gain A can be used between the auxiliary outputs

of the HP141T and the input channels of the [ADC](#). To shift the output voltages, a logic level-shifter can be used to translate the signals to be within the operating voltage domain.

Sampling Limitations in the Data Acquisition and Data Processing Subsystems

Among the constraints in the selection of the [ADC](#) is the requirement of the data acquisition subsystem to sample 801 points. Given that the 8552B has 16 internal scan rates from 0.1 ms/div to 10 s/div in a 1, 2, 5 sequence and given that the data acquisition subsystem needs to store 801 points per scan, the fastest scan rate, corresponding to 0.1 ms/div, can be achieved using an [ADC](#) with a sampling rate of

$$\frac{801 \text{ samples}}{0.1 \text{ ms}} = 8.01 \text{ MSPS}$$

Similarly, for the slowest scan rate corresponding to 10 s/div, the required minimum sampling rate of the [ADC](#) is 80.1 [SPS](#).

In addition to the sample rate of the [ADC](#), the design must take several factors into account regarding the integration of the [ADC](#) with a digital processor such as a [SBC](#), [MCU](#), or [FPGA](#). Table 3.4 shows the considerations that need to be made in selecting a digital processor that is compatible with the operation of the [ADC](#) and high sampling rate.

Digital Processor Type	Consideration	Feature
MCU	Communication Interface	Supports I ² C and SPI
	Data Transfer Rate	Up to hundreds of MHz
	Synchronization and Timing	Can use interrupts and FIFO buffers
SBC	Communication interface	Supports I ² C and SPI
	Data Transfer Rate	Suited for low sampling rates
	Synchronization and Timing	Requires buffering and introduces operating system relating delays
FPGA	Communication Interface	Supports SPI and high-speed parallel interfaces
	Data Transfer Rate	Can handle sampling rates at hundreds of MHz to GHz speeds.
	Synchronization and Timing	Uses parallel data pipelines suitable for high-speed sampling rates for real-time spectral analysis applications.

Table 3.4: Digital processor selection based on [ADC](#) interface considerations.

Table 3.4 indicates that to fulfil the requirement for sampling 801 points per scan, the design that can deliver the highest performance in real-time data acquisition should include an [ADC](#) which can perform 8.01 [MSPS](#) and a [FPGA](#) with multiple pipelines for storing and processing rapidly sampled data in parallel. Alternatively, a heterogeneous digital processor can be implemented using a suitable combination of the available options listed in the table. For example, a Xilinx Artix-7 [FPGA](#) can be used for high-speed data acquisition while a Raspberry Pi [SBC](#) is used to process and display data. [MCUs](#) can also be daisy-chained to perform different tasks in the data acquisition subsystem.

Further Design Considerations in Digital Processor Selection

While **FPGAs** offer superior speeds in data acquisition and processing, one of the shortcomings of selecting it as the primary digital processor is that the overall development time is typically longer than the design time for **MCUs** and **SBCs**. This is because **FPGA** development has a structured design flow which includes details of the register-transfer level (**RTL**) using a hardware description language (**HDL**), validation, synthesis and implementation. Although tools like AMD Vivado exist to significantly reduce **FPGA** development times, the other available options offer significantly shorter development times.

Table ?? gives a comparison between the available digital processor options to highlight the most suitable option for fulfilling the requirements of this project.

Design Aspect	MCU	SBC	FPGA
Development Time	Short	Moderate	Long
Processing Speed	Medium	Moderate to High	High
Programming Language	C/C++	Python or C/C++	VHDL or Verilog
On-board Memory and RAM	Very Low	High	Low
Power consumption	Low	High	Moderate to High
Scalability	High	Moderate	High
Cost	Low	Moderate to High	High

Table 3.5: A structure comparison of the available digital processor options.

The different benefits of choosing a certain digital processor can be deduced from Table 3.4 and ?? . Other considerations need to be made regarding the memory capacity of the chosen processor, however, the primary objective is to satisfy UR06 which requires the data acquisition subsystem to be capable of storing and recalling traces. Due to the high sampling rate, the device must have a sufficient large memory capacity for storing signal frequency and amplitude related information at high-speeds. Specifically, given that a minimum sampling rate of 8.01 **MSPS** is required and supposing that the resolution of the **ADC** is 16-bit, the device must be able to store

$$8.01\text{MSPS} \times 16 \text{ bit} = 128\,160\,000 \text{ bit s}^{-1}$$

In other words, the device must be able to store 16.02 MB s^{-1} , which equates to approximately 1 GB per minutes. Thus, the selected digital processor must include efficient memory management algorithms to ensure that data is not lost or corrupted. The table indicates that the **MCU** has the most limited capability with respect to memory. This is because **MCUs** typically have few KB to MB of **RAM** and require external memory. Similarly, **FPGAs** offer limited memory capabilities since they typically use a small embedded **RAM** and require external memory for large data buffers.

In summary, the **MCU** is most suitable for its low cost, low power consumption, good real-time capabilities and short development time. However, the **MCU** is only suitable for moderate-speed sampling, which is not ideal for the high sampling rates that are needed to fulfil the requirements. While **SBCs** offer intermediate performs and features that combine the moderate capabilities of **MCUs**

and **FPGAs**, their primary shortcoming is poor real-time capabilities due to **OS** delays and overall low latency. Finally, the best option for this application is the **FPGA** due to its ability to perform real-time high-speed data processing and **ADC** synchronization, however, the biggest shortcomings are the development time and high cost.

Due to time constraints and low availability of the STM32H723ZG development board, inquiries into alternative devices were taken into consideration to ensure that data can be sampled in the case where the STM32H723ZG is not available. The Field-Fox is a spectrum analyzer that can be used to acquire amplitude vs frequency data directly, similar to the HP141T system. However, to extract amplitude vs frequency plot data in the time domain corresponding to the time series outputs of the HP141T **SA**, a digital oscilloscope can be used. This includes devices such as the R&S RTC1000 oscilloscope Rhode & Schwarz and Keysight HD304MSO with 4 analog channels. Most modern digital oscilloscopes can measure various parameters like voltage, frequency and phase, and the ability to interface with a personal computer. Sizes of these oscilloscopes can range from small and portable to very large devices that are typically used in lab settings. Smaller, hand-held devices include products from Fluke, Pico Technologies, and Tektronix.

These oscilloscopes can be used as alternative sampling devices, given that the device is capable of storing data in a readable format. Digital oscilloscopes typically have an associated desktop software which can display samples of channels linked to input connects. However, given that the aim of the project is to digitize the HP141T **SA** in accordance with the tabulated requirements, the **GUI** that comes with an oscilloscope cannot be used as the graphical interface of the design because it may not necessarily fulfil the requirements. Instead, the oscilloscopes and accompanying software can be used as alternative sampling devices in the data acquisition subsystem, or for taking measurements during the testing and verification phase of the design process.

3.3.3 Design Considerations in Satisfying Display Requirements

The choice of display depends on the selected processor type. The display hardware must satisfy multiple user requirements including UR03, UR07 and UR09 which are related to resolution, functionality and power. Further considerations have to be made for the display to satisfy requirements relating to the user interface (**UI**). The following section breaks down these requirements separately to evaluate the design choices that need to be made in modernizing the HP141T display.

Display Hardware Considerations for a **MCU**-Based System

Table 3.6 assimilates the design considerations for a **MCU**-based system. Overall, the design needs to consider factors such as clock and I/O speed, memory, power consumption and the display interface.

Display Hardware Considerations for a **SBC**-Based System

If the chosen digital processor is a single-board computer such as the Raspberry Pi 4B, similar considerations need to be made regarding its compatibility of the display unit. Differences arise from the hardware capabilities that the **SBC** offers. Table 3.7 summarizes the design considerations that need to be made for the interface between **SBC** and the display unit.

Key Consideration	Aspects	Description
Processor	Clock Speed	Clock speed of at least 100 MHz for rendering 801 points per scan to the display
	Architecture	A 32-bit ARM Cortex-M series MCU is preferred for good balance between power efficiency and speed
	I/O Speed	ADC waveform samples need to be rendered at high speeds. SPI is recommended for improving latency
Memory	RAM	At least 128 kB of RAM required for real-time display rendering
	Flash Memory	Required for storing firmware and display configurations which need to be loaded during power-up.
Interface	Compatibility	Should support the same communication interfaces as the display screen.
	Resources	Modern displays include internal frame buffers which can reduce the effect of memory and data transfer bottlenecks.
Power	Compatibility	System must be powered by a single wall wart power supply.
	Efficiency	Low-power and standby modes can reduce energy consumption when the display is idle.

Table 3.6: Key considerations for the display subsystem for a [MCU](#)-based system.

Key Consideration	Aspects	Description
Resolution	1080p at 60Hz	Recommended for smooth spectrum rendering; higher resolutions.
	HDMI 2.0	A high-speed HDMI 2.0-compatible display is required for the best display quality.
HDMI	HDMI Version Support	Older monitors may only support HDMI 1.2, which could limit resolution and refresh rates
	Data Handling	Improper extended display identification data detection may require manual configuration
	Adapter	Micro-HDMI to HDMI adapters are required. Faulty adapters may introduce noise
Touch	Touch Display	Dedicated SBC touchscreens can be used
	DSI	Raspberry Pi touchscreen connect via the DSI (Display Serial Interface) without required HDMI
	Power	Dedicated screen can be powered by the SBC
Monitor	Compatibility	Selected display must support the required resolution and refresh rate for clear visualization of digitized spectra.
	Over-scanning	Display adjustments may be required for screens that crop spectrum visualizations.
Software	OS Support	Dedicated OS recommended for compatibility with built-in display drivers and utilities.
	Settings	Used to adjust display resolution, rotation, and other settings for the most suitable spectrum visualization.

Table 3.7: Display considerations for screen integration with a [SBC](#)-based system.

Display Hardware Considerations for a **FPGA**-Based System

Although **FPGAs** offer the best performance with respect to sampling rates and data processing speeds, they are not as suitable for processing display data compared to **SBCs**. However, the real-time capabilities of the **FPGA**-based system can provide a seamless user experience by rendering spectrum displays at a high frame rate. Table 3.8 lists the design considerations for the display of a **FPGA**-based system.

Key Consideration	Aspects	Description
Video Output	HDMI Support	Different FPGAs feature built-in HDMI , DVI transmitters or external serializer chips. The chosen FPGA must support high-speed video output
	Parallel	For parallel RGB inputs, the FPGA must generate compatible timing signals for proper synchronization
	Buffering	Additional external memory may be required for storing video frames before display
Synchronization	Pixel Clock	Display's resolution and refresh rate requirements must be compatible with a stable FPGA pixel clock
	Fabric Speed	Spectrum updates can be rendered rapidly using the high-speed fabric which ensure low-latency
Interface	Protocols	The display must be able to interface with the available hardware on the FPGA
	Controller	Display must be compatible with FPGA IP cores for developing display functionality more easily
	Resources	Optimization is needed to balance consumption of FPGA logic, memory block and I/O pins by the display unit
Memory	RAM	External DDR memory may be necessary for extending limited on-chip RAM capacity
	Storage	Efficient memory management is essential to prevent bottlenecks
Power	Delivery	Some high-speed video interfaces require additional power regulators
	Heating	FPGA video processing generates heat which may require cooling to prevent damage to components of the FPGA
Software	Drivers	Display drivers must be implemented using HDL or software cores
	Updates	Updates must occur in real-time and support different display modes

Table 3.8: Display considerations for an **FPGA**-based HP141T modernization system.

Display User Interface

User requirements relating to the aspects of the display that the user can interact with include UR02, UR04, UR05 and UR07. The primary objective of the project is to display a plot of the amplitude vs frequency for the input signal, in real-time. To achieve this objective, spectrum data needs to be

organized and easily accessible from [RAM](#) or flash memory, depending on the architecture of the system that is centred around the [ADC](#) and digital processor. For example, STM microcontrollers have a dedicated [LCD-TFT](#) display controller with a frame buffer that can be located either in on-chip memory or in an external memory, depending on the resolution. Using external memory can increase latency which decreases the real-time performance of the system.

In addition, data needs to be scaled while considering the effects of windowing, so that all spectrum data can be translated to a grid with 8 divisions of the amplitude, on the vertical axis, and 10 divisions of the frequency, on the horizontal axis. This is also to ensure that the graphical interface resembles the appearance of the [CRT](#) display, as shown in Figure 3.4, is maintained.



Figure 3.4: HP141T [CRT](#) display section.

Secondary objectives include digitizing the operations that were previously executed by physical knobs such as horizontal and vertical scaling of the waveform. This objective can be as functions in a program that can be called to perform similar task. The choice of coding language and available development tools for the program that runs the display is a critical part of the [UI](#) subsystem.

An alternative approach to handling data acquisition and display functions on the same board would be to use a single-board computer for the display. Spectrum data can be retrieved from memory and displayed to the user in a Python, C/C++ or Java program. Knobs can be replaced by programmatic buttons with functions that users can access by touching the screen. This can allow other features such as averaging and displaying maxima and minima to be easily accessible to the user. For this project, [UI](#) must enable users to select between different display modes, including the [PHM](#), [AvM](#) and [RwM](#). The [UI](#) must enable users to switch between these modes seamlessly and allow for back them to return to previous screens using a back button in the program.

Overall, the project aims to adhere to Nielsen's heuristic design principles which includes:

1. Match between the system and the HP141T - the program should use the language and current features of the HP141T that users are familiar with

2. User control and freedom - the **UI** must support undo, redo and exit points
3. Aesthetics and minimalist design - the most relevant information that must be displayed is related to the frequency plot of the input signals as well as configurations and settings of the data acquisition and digital processing units that give context to the current display
4. Flexibility and efficiency of use - the **UI** interface of the modernized HP141T must be optimized for difference experience levels through suitable shortcuts and frequent actions such as averaging and showing maxima and minima of the input signal.
5. Help and documentation - documentation of the modernized HP141T display must correspond to current documentations of the entire system and must be easy to read
6. Error prevention - the **UI** must protect users from deleting stored traces and signal information
7. Visibility - a user must always know what is happening within the program and current state of their actions
8. Consistency and standards - the appearance of the **UI** must follow the conventions of modern spectrum analyzer displays through consistent words, situations and actions.
9. Help user to recognize, diagnose and recover from errors such as closing windows with information about pulses or signal noise
10. Recognition rather than recall - minimize users' memory load by making program features that are related to the most accessed signal data easy to access

3.3.4 Equipment for Debugging and Testing

The following section gives a review of the project requirement which necessitates the development of an HP141T display emulator of the three auxiliary outputs as shown in [??](#). Additionally, the section describes software-based simulation tools that can be used in the development for debugging and obtaining an idea about the expected behaviour individual hardware components and overall system.

Emulating HP141T Auxiliary Outputs

The sawtooth signal can be generated using a 555-timer circuit which produces a ramp voltage with adjustable frequency to mimic the adjustable scan time, and a fixed peak-to-peak voltage of 10 V with no **DC** offset. This is to ensure that the horizontal sawtooth output range between -5 V and $+5\text{ V}$ is maintained.

To emulate the vertical output which swings between 0 V and -0.8 V , a simple voltage biasing circuit can be built with appropriate resistor values and input voltage. To control this level, a potentiometer can be used to manually adjust the voltage level within that range. Similarly, the pen-lift output can be produced using a biasing circuit with a potentiometer that allows the voltage to swing between 0 V and 14 V .

Lastly, the HP141T display emulator circuit needs to interface with the signal conditioning circuit that prepares the signal voltages to match the operating range of the **ADC**. Therefore, to avoid the

development of multiple circuits, the output cables of the display emulator must be the same as the output cables that take the auxiliary outputs from the HP141T to the signal conditioning circuit.

Software-Based Simulation Tools

To optimize the selection and configuration of the data acquisition subsystem, [ADCs](#) can be simulated in MATLAB using Simulink to understand the behaviour of the sawtooth input from the HP141T or emulating hardware. Additionally, Simulink provides visualizations of the spectrum of the [ADC](#) output which can be used to determine the expected values from sampling signals. Simulink also allows users to add impairments to the [ADC](#) for introducing a layer of non-linearity that may be seen for signals with frequencies that are close the edges of the [RBW](#) or intermediate frequency.

Electronic circuits, such as the signal conditioning circuits with op-amps and level-shifters, can be simulated using LTSpice to predict the expected behaviour. Output values from LTSpice simulations can be used to simulate the behaviour of the system using Simulink with MATLAB.

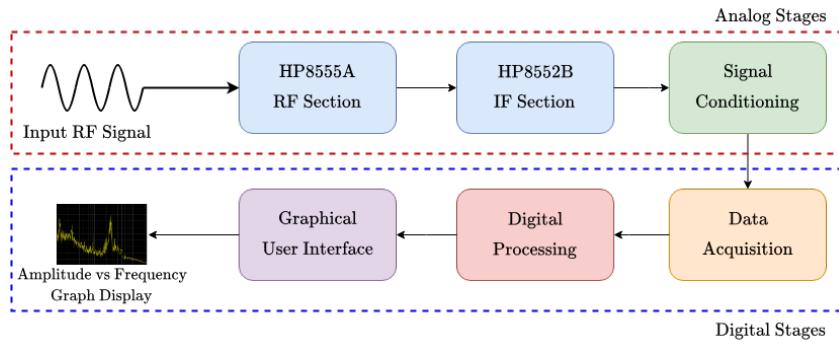


Figure 3.5: Overview system block diagram for the modernized HP141T which employs the HP8555A and HP8552B plug-in sections.

3.4 System Design

Design decisions in formulating the specifications were informed by the requirement review in the previous section. This section aims to give a full description of the system and the interfaces between the different subsystems. The section begins with a visual representation of the system using a block diagram. Following the visual representation of the design, the section includes specifications and concludes with acceptance and integration tests.

3.4.1 Overview System Block Diagram

The system block diagram shown in Figure 3.5 below is a graphical overview of the different stages that an RF signal goes through before the amplitude vs frequency graph is displayed. The modernized HP141T maintains the basic analog signal processing functionality performed by the HP8555A **RF** section and the HP8552B **IF** sections. The scope of the proposed design does not involve any changes to the signal processing components of the plug-in sections. However, the Signal Conditioning Subsystem (**SCS**) prepares the three auxiliary outputs from the HP552B **IF** section for the Data Acquisition Subsystem (**DAS**) which requires input voltages in the range between 0 V and 3.3 V.

ADCs in the **DAS** play a primary role in digitizing the HP141T system by converting the conditioned analog signals into digital values which can be processed in the Digital Processing Subsystem (**DPS**) which enables manipulation of the digital values. This is particularly necessary for averaging and executing other arithmetic operations in order to satisfy the needs of the user. The Graphical User Interface Subsystem (**GUIS**) manages the display and user preferences for displaying the amplitude vs frequency information of the input **RF** signal.

Overall, the can be modularized into five primary subsystems, including:

1. HP141T Subsystem
2. Signal Conditioning (**SCS**)
3. Data Acquisition (**DAS**)
4. Digital Processing (**DPS**)
5. Graphical User Inferface (**GUIS**)

Although this project does not involve the redesign of the HP141T subsystem, an HP141T Emulator design is proposed for simulating the outputs of the IF plug-in section. In particular, the emulator produces signals that are similar to the horizontal, vertical and pen lift outputs of the HP8552B plug-in. This implies that the emulator can be considered as a secondary subsystem that can substitute the HP141T subsystem during calibration of the digital section of the proposed design, as illustrated in figure 3.6. Furthermore, the diagram illustrates that the emulator does not have high frequency input signals like the HP8555A RF section plug-in. Instead, oscillators and function generator chips are proposed for producing outputs that approximate the auxiliary outputs, as shown in table ??.

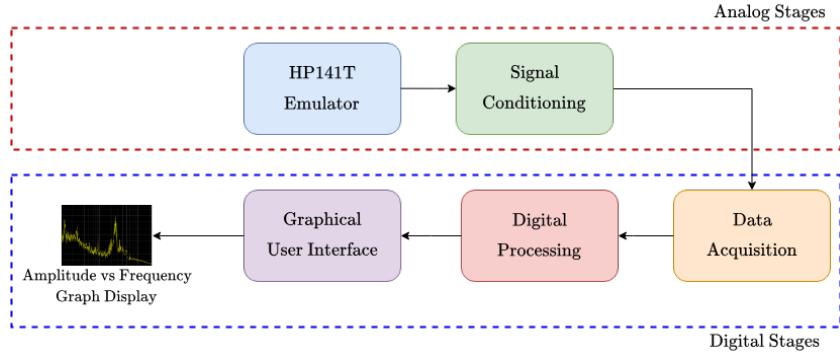


Figure 3.6: Calibration system diagram where the HP141T Emulator is used to configure the primary subsystems such as the SCS, DAS, etc.

All analog stages are executed in PCBs, and the digital stages are performed by a microcontroller and single-board computer. The following sections detail the specifications of the system modularization and components.

3.4.2 System Modularization

The system block diagram is expanded as illustrated in figure 3.7 where the primary components of each of the subsystems are shown. In the diagram, arrows with a solid line illustrate interfaces between subsystems, while arrows with dotted lines describes the flow of analog or digital signals within each subsystem.

Figure 3.7 is accompanied by table ?? showing subsystem components in more detail. The table also

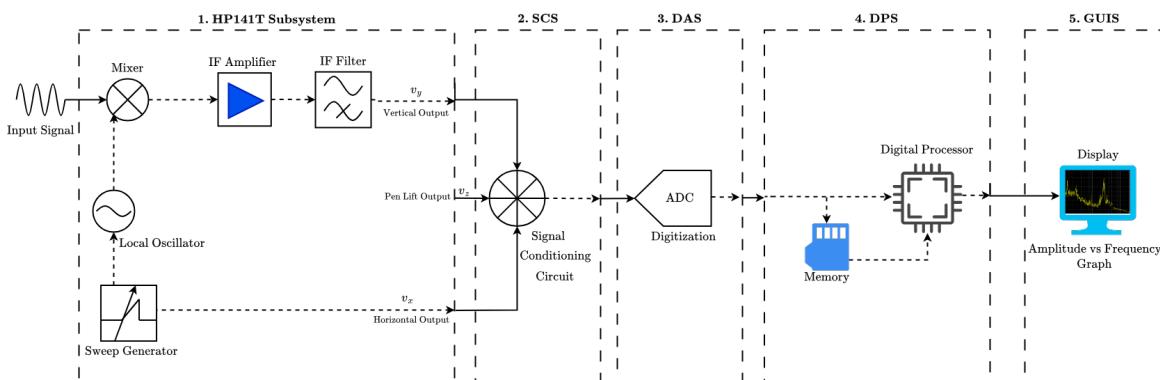


Figure 3.7: Expansion of the overall system diagram showing the primary components and functions in each subsystem.

Subsystem	Component	Signal Type
HP141T Subsystem	HP8555A	Analog
	HP8552B	Analog
HP141T Emulator	Vertical Output Simulator	Analog
	Horizontal Output Simulator	Analog
	Pen Lift Output Simulator	Analog
Signal Conditioning Subsystem	Op-amp based vertical output conditioning circuit	Analog
	Op-amp based horizontal output conditioning circuit	Analog
	Op-amp based pen-lift output conditioning circuit	Analog
Data Acquisition Subsystem	ADC	Analog/Digital
	MCU	Digital
	Storage Device	Digital
Digital Processing Subsystem	MCU	Digital
	SBC	Digital
Graphical User Interface Subsystem	SBC	Digital
	LCD Touch Screen	Digital/Analog
	Software library	Digital

Table 3.9: Showing components in each subsystem and the type of signals that are processed by each unit.

details the components of the HP141T Emulator subsystem. Signal types are noted in table ?? because a significant part of the system behaviour can be attributed to the interchange between continuous or discrete values.

HP141T Subsystem Specifications

Table ?? shows the expected outputs collected from the technical manual of the plug-in which also details the expected frequency range.

Output	Symbol	Voltage Range	Bandwidth
Horizontal	V_x	-5.0 V to 5.0 V	0.005 Hz to 500 Hz
Vertical	V_y	-0.8 V to 0 V	10 Hz to 300 kHz
Pen Lift	V_z	0 V to 14 V	0 Hz

Table 3.10: Characteristic values of the HP8552B IF plug-in section's analog output signals [24].

Table ?? shows the amplitude specifications of the HP8552B plug-in section.

Operation Mode	Amplitude Calibration Range	Amplitude per Division (dB/div)	Display Range (dB)
Logarithmic	-130 dBm to +10 dBm	10	70
Linear	-23 dB V to +117 dB V	10	70

Table 3.11: Amplitude specifications of HP8552B plug-in section [24].

The HP8552B has four operational scan modes as shown in table ???. The table also shows the scan time specifications of the HP8552B.

Scan Characteristic	Description
Scan Time	16 scan rates from 0.1 ms/div to 10 s/div in a 1, 2, 5 sequence
Scan Time Accuracy	$\pm 10\%$ for scan rates from 0.1 ms/div to 20 ms/div $\pm 20\%$ for scan rates from 50 ms/div to 10 s/div
Scan Mode	<i>Internal mode</i> for repetitive scanning by internally generated ramp <i>Single mode</i> actuated by panel push button <i>External mode</i> determined by a 0 to +8 V external signal <i>Manual mode</i> controls the scan by the position of the manual knob
Scan Trigger	<i>Auto trigger</i> runs the scan freely <i>Line trigger</i> synchronizes the scan with the power line frequency <i>External trigger</i> synchronizes the scan with an external signal (20 V max) <i>Video trigger</i> synchronizes the scan to an envelope of the RF input signal

Table 3.12: Scan time and scan mode specifications of the HP8552B plug-in [24].

HP141T Emulator Subsystem Design

According to UR10, the outputs of the HP141T emulator subsystem must simulate the three outputs of the HP8552B **IF** plug-in section as described in table 1.1. To satisfy this requirement, the HP141T emulator subsystem follows the system diagram illustrated in figure 3.8 where a XR2206 is used as a function generator for producing the vertical output, sawtooth scan output corresponding to the horizontal output, and the pen up or pen down state of the pen-lift output.

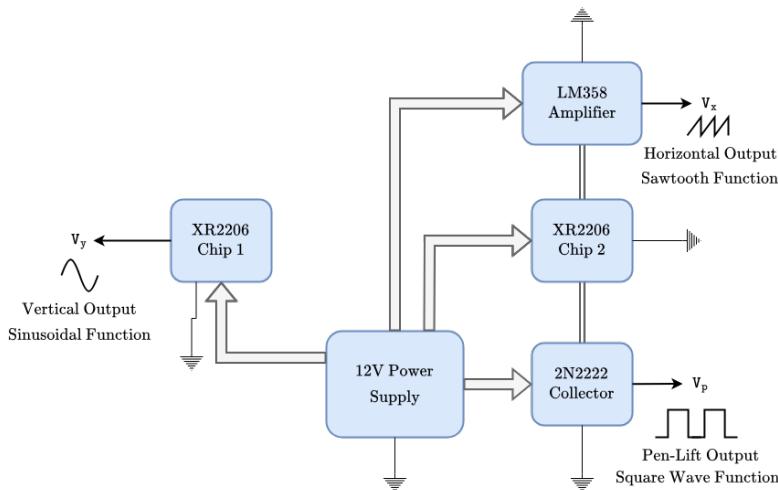


Figure 3.8: The HP141T Emulator subsystem diagram showing the primary components powered by the same ± 12 V DC dual rail supply.

The frequency range of the -0.8 V– 0 V vertical output, V_y , is limited to the bandwidth of the video filter in the **IF** section, i.e. 10 Hz to 300 kHz [24]. The front panel check procedure in the technical manual of the HP8552B **IF** section sets a frequency range of up to 10 kHz in the video filter, corresponding to a 30 MHz fundamental signal at the input of the **RF** section plug-in [18].

Two XR2206 functional generator chips are implemented for emulating the HP8552B outputs, V_y

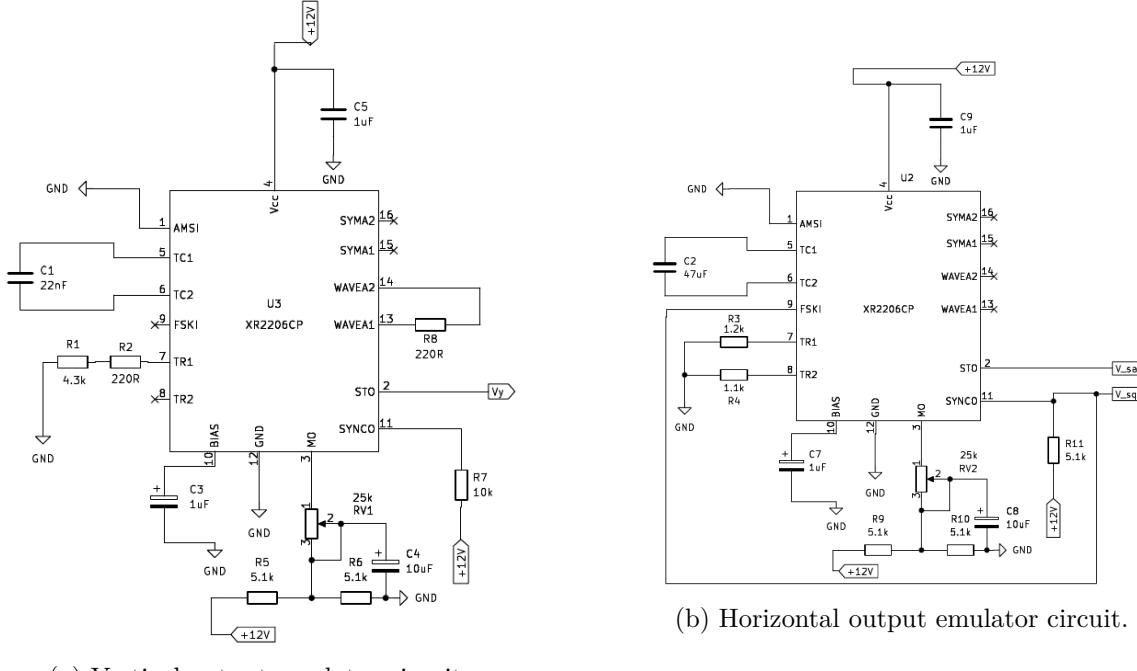


Figure 3.9: Emulator circuits for vertical and horizontal outputs of the HP8552B IF section.

and V_x . For V_y , timing resistors and capacitors are employed to emulate video filter output during the configuration procedure described in the chip's datasheet which implements a sinusoidal wave with a calibration frequency of 10 kHz, corresponding to a 30 MHz RF signal at the input of the HP8555A plug-in. The full vertical output simulator circuit is illustrated in figure 3.9a, adopted from the recommended configuration [25]. The expected shape and time domain characteristics of the 3 V_{pp} sinusoidal wave are shown in figure 3.10.

The required frequency is produced by an internal voltage-controlled oscillator that is set by a timing resistor at pin 7 and 8. Pin descriptions and the values of the capacitors and resistors that are connected to them are recorded in table 3.13 and 3.14.

The amplitude of V_y and V_x at the output of the XR2206 chips is controlled by adjusting the variable resistors RV1 and RV2 and limited to 0. To generate a sinusoid with a constant peak-to-peak voltage

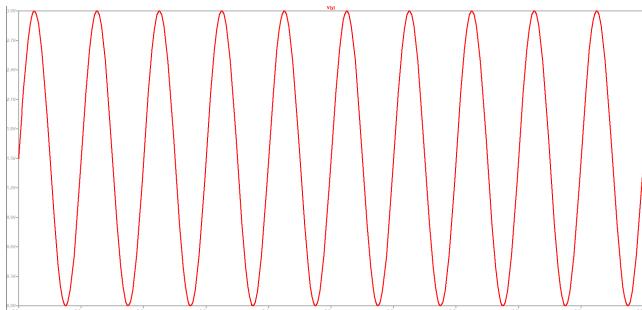


Figure 3.10: Simulated output at the STO pin of the XR2206 chip employed in emulating the vertical output of as a 3 V_{pp} sinusoidal wave.

and constant frequency for the vertical output, an external capacitance of $C = C_1$ is connected across

Table 3.13: Pin connection description for the XR2206-based vertical output simulation circuit.

Pin No.	Type	Symbol	Description
1	I	AMSI	A 0 V amplitude modulation signal input.
2	O	STO	Sinusoid wave output.
3	O	MO	Multiplier output for controlling the amplitude of V_y
4	I	V_{CC}	+12 V power supply with decoupling capacitor.
5	I	TC1	Timing capacitor.
6	I	TC2	Timing capacitor.
7	O	TR1	Timing resistor.
8	X	TR2	No connection.
9	X	FSKI	No connection.
10	O	BIAS	Internal voltage reference.
11	O	SYNCO	Open collector with a pull-up resistor to the power supply.
12		GND	Ground pin.
13	I	WAVEA1	First waveform-adjustment input.
14	I	WAVEA2	Second waveform-adjustment input.
15	X	SYMA1	No connection for wave symmetry adjustment.
16	X	SYMA2	No connection for wave symmetry adjustment.

pin 5 and 6, and a timing resistance $R = R_1 + R_2$ across pin 7, such that the operational frequency f_y of the XR2206 is

$$f_y = \frac{1}{RC} \text{ [Hz]} \quad (3.1)$$

Similarly, the characteristics of the horizontal sawtooth output depend on the timing components at pin 5, 6, and 7, as illustrated in figure 3.9b. However, when operating as a sawtooth function generator as in the case of the horizontal output emulator circuit, the frequency f_x is given by,

$$f_x = \frac{2}{C(R_a + R_b)} \text{ [Hz]} \quad (3.2)$$

where R_a and R_b correspond to the timing resistors connected to pin 7 and pin 8, and C is the timing capacitor connected to pin 5 and pin 6, respectively. The duty cycle of the output sawtooth function is given by the parallel combination of R_a and R_b , such that

$$\text{Duty Cycle} = \frac{R_a}{R_a + R_b} \quad (3.3)$$

The design associates a duty cycle with the proportion of time that the signal spends rising from the minimum voltage to the maximum of the sawtooth function in the output of the emulator, i.e. as the XR2206 STO output voltage increases from the 0 V to 3 V. The design approximates the rise time, t_{rise} , and duty cycle of this output from the manual of the HP8552B as illustrated in figure 3.11 [24]. Using $R_a = R_3 = 1.1 \text{ k}\Omega$ and $R_b = R_4 = 1.2 \text{ k}\Omega$, the duty cycle is approximated to 52.2% and the rise

time to 54 ms. These time domain characteristics correspond to a horizontal oscillation frequency of 18.52 s, given that the timing capacitance is $47 \mu\text{F}$.

Table 3.14: HP141T emulator electrical specifications.

Designation	Parameter	Minimum	Maximum	Units	Conditions
General Characteristics					
V_{IN}	Split Supply Voltage	9	12	V	
I_{IN}	Supply Current	12	17	mA	
Output Voltage Characteristics					
V_y	Vertical Output	0	3	V	$RV1 = 25 \text{ k}\Omega$
V_x	Horizontal Output	-4.75	4.76	V	$RV2 = 15 \text{ k}\Omega$
V_p	Pen Lift Output	0	12	V	
Frequency and Time Domain Characteristics					
f_y	Vertical Output Oscillation Frequency		10	kHz	$R_1 = 4.3 \text{ k}\Omega, R_2 = 220 \Omega, C_1 = 22 \mu\text{F}$
f_x	Horizontal Output Frequency	17.24	20	Hz	$R_3 = 1.2 \text{ k}\Omega, R_4 = 1.3 \Omega, C_2 = 47 \mu\text{F}$
t_{rise}	Horizontal Output Sawtooth Rise Time	50	58	ms	$R_3 = 1.2 \text{ k}\Omega, R_4 = 1.3 \Omega, C_2 = 47 \mu\text{F}$
T_x	Sawtooth Period	95.6	111	ms	

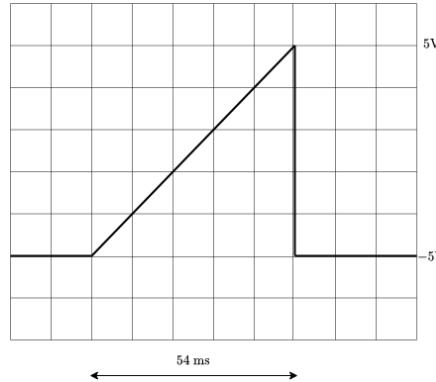


Figure 3.11: Illustration of the duty cycle and rise time of sawtooth function corresponding to the horizontal output, V_x , of the HP8552B.

Since the output voltage of the multiplier and wave shaper in the XR2206 is limited between 0 V to 3 V, the circuit in figure 3.12 is used to adjust the sawtooth output from the chip to rise from -5 V to +5 V.

The sawtooth output of the XR2206 circuit is simulated in LTSpice using the `PWL` command as shown in figure 3.13, with the corresponding results of the simulation shown in the figure 3.14. The `PWL` command simulates a piecewise function between time t_1 and t_n , $n \geq 2$. The simulation shows that the expected voltage range of the scan output lies between -4.75 V and 4.76 V with a steep rise time and a clipped voltage at 4.76 V. The simulation also indicates a start time delay in the output. This is due to the capacitors in the LM358 op-amp which introduce RC time constants during discharge. This delay, steep rise time, and voltage clipping of the sawtooth is accounted for and adjustment are made in software, as specified in the following sections.

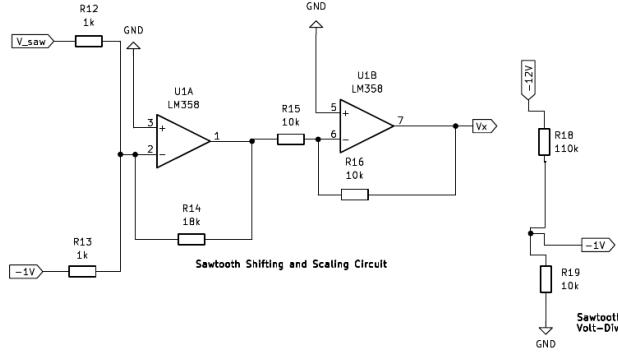


Figure 3.12: XR2206 sawtooth output level shifter for achieving the appropriate voltage range.

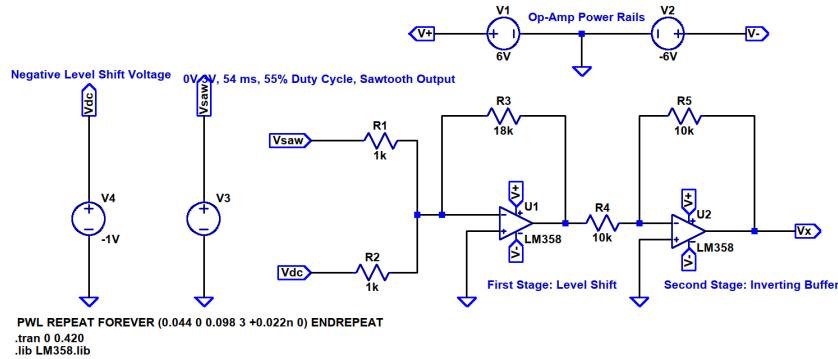


Figure 3.13: Simulated level shift circuit.

The first stage of the level shifting and scaling circuit in figure 3.12 and 3.13 using an LM358 op-amp as a difference amplifier with a gain of 18 to scale the sawtooth voltage, in the range between 0 V and 3 V, to the desired range from -5 V to $+5 \text{ V}$. The second stage is an inverting buffer which serves to invert the output of the first stage such that the gradient of the sawtooth increases in the same direction.

The period in the simulated sawtooth remains the same, however, the duty cycle decreases due to the delay and shorter rise time of the output, as seen in figure 3.14. The duty cycle is scaled from 52.2% to 40.9% while the start time of the output is delayed by 18 ms and the rise time is scaled from 54 ms to 10 ms, indicating a time scaling factor of 5.4. The start time of the output signal is delayed by 13 ms from the start time of the XR2206 STO output. The software accounts for the time scaling, delay and

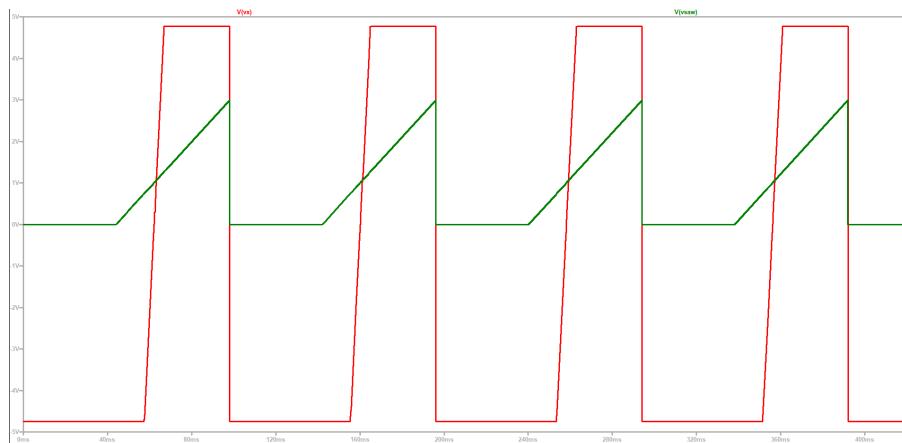


Figure 3.14: Showing simulated scan output of the emulator.

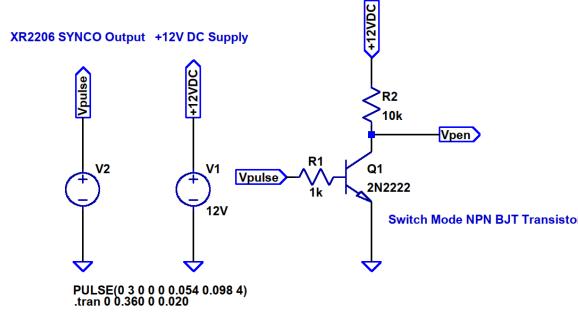


Figure 3.15: Simulation of the pen lift output using the XR2206 SYNCO pulse output.

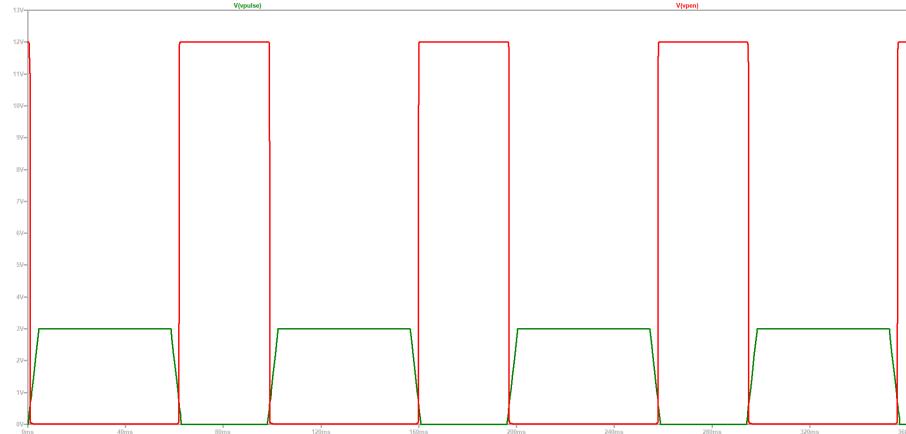


Figure 3.16: Pen lift output simulation with the XR2206 SYNCO pulse in green and the pulse at the collector of the NPN transistor in red.

change in the duty cycle to ensure accurate simulation of the rise time of 54 ms.

The states of the pen-lift output are produced using a square wave pulse at the SYNCO pin output of the XR2206 chip. The SYNCO pulse has the same duty cycle and period as the sawtooth output at the STO pin. Ideally, this allows for the design to synchronise the pen-lift output states with the scan time. A 2N2222 NPN transistors is employed as a switch such that the collector voltage goes high (12 V) when the SYNCO pulse is off, as shown in figure 3.16. In other words, when the collector output is high, the pen-lift output is in the `pen down` state. Conversely, when the collector output is low, the pen-lift output is in the `pen up` state. The implications of these states on the display are discussed in detail in section ??.

Designation	Parameter	Minimum	Maximum	Units	Condition
V _B	Base input voltage	0	3	V	R _B = 1 kΩ
V _C	Collector output voltage	0	12	V	R _C = 10 kΩ
V _E	Emitter voltage	0		V	

Table 3.15: Specifications of the pen lift circuit as part of the HP141T emulator subsystem based on the 2N2222 NPN transistor.

A simulation of the XR2206 SYNCO square wave and pen-lift output is shown in figure 3.15 where the PULSE command is used in LTSpice. The base and collector voltages of the NPN transistor, and other specifications, are included in table ???. The table corresponds to the simulation output where the transistor turns on and pulls the collector voltage to 0 V when the SYNCO square wave is high

(3 V). When the square wave is low (0 V), the transistor turns off and the collector output is pulled up to 12 V through the $10\text{ k}\Omega$ resistor.

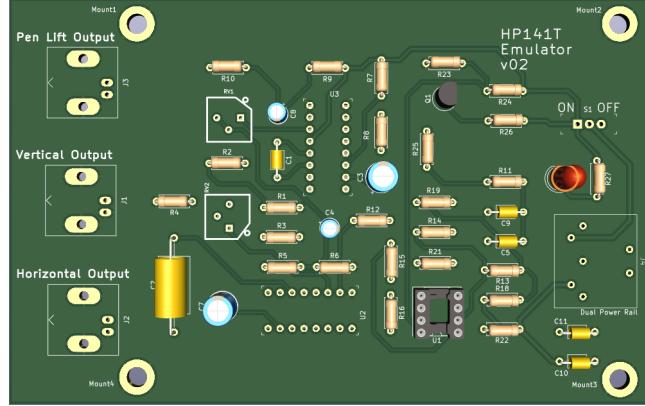


Figure 3.17: 3D view of the HP141T Emulator subsystem's PCB.

The full HP141T Emulator subsystem is implemented on a PCB as illustrated in the 3D model in figure 3.17. The output of the system is interfaced through a BNC coaxial cable for its low loss capabilities for improved performance.

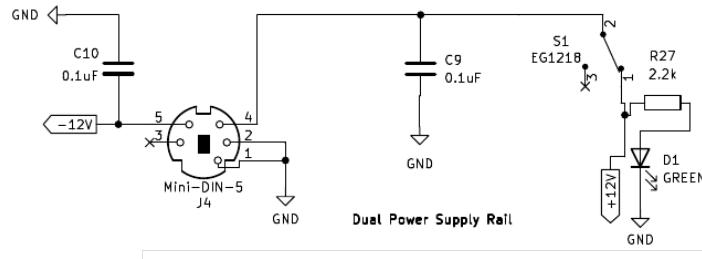


Figure 3.18: HP141T Emulator power supply circuit with decoupling capacitors to reduce high frequency noise.

A 28 W power brick that converts 230 V, 50 Hz AC to ± 12 V DC dual supply. The power supply is connected to the PCB through a 5 pole din socket as illustrated in figure 3.18. Decoupling capacitors are used to block high frequency noise by creating a high-pass RC filter with the total impedance of the subsystem that smooths any fluctuations in the power supply. The subsystem can be switched on by setting the 3 pole switch to the left-hand position and switched off by setting it to the centre right-hand position. A power LED is used to indicate whether the subsystem is on or off.

Signal Conditioning Subsystem Design

The SCS forms the central part in the digitization and modernization of the HP141T SA as it interfaces with the HP141T subsystem and HP141T Emulator subsystem through three separate BNC coaxial cables and prepares the horizontal, vertical and pen-lift outputs to a suitable input voltage range for the unipolar operation STM32H723ZG board, i.e. 0 V to 3.3 V DC. This implies that the outputs of the HP8552B IF section or HP141T Emulator are the inputs to the SCS at the BNC ports illustrated in figure 3.19 showing the circuit schematic of the BNC connector which takes in the vertical output V_y .

Conversion of the input voltages to the appropriate level of 3.3 V for the analog-to-digital interface

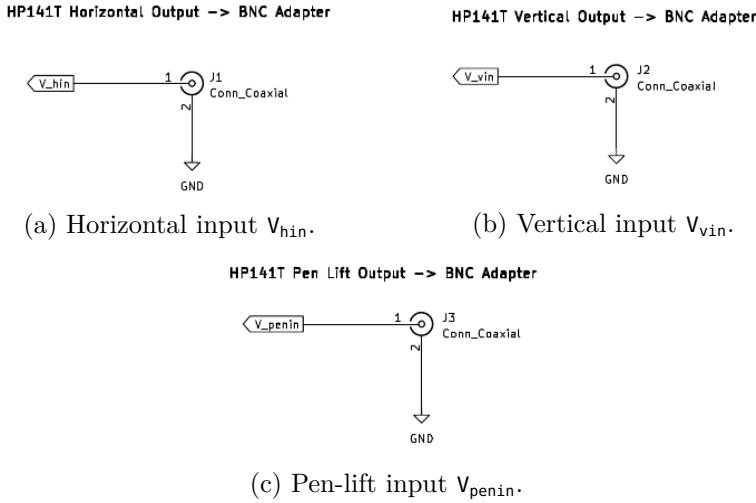


Figure 3.19: Schematic of the input connectors that enable the **SCS** to interface with the outputs from the HP8552B plug-in or HP141T Emulator subsystem.

employs three LM358 op-amps in different pipeline circuits that apply the required scaling and level shifting for each input. The flow of each input through this pipelined analog subsystem is illustrated in figure 3.20. The circuits in each pipeline share a common GND to reduce noise in the system and ensure that the **ADC** in the **DAS** produces accurate results.

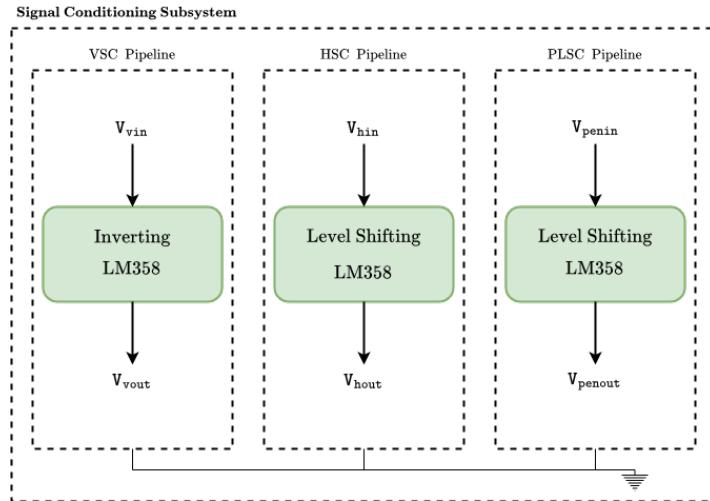


Figure 3.20: **Signal Conditioning Subsystem** diagram.

In the **Horizontal Signal Conditioning (HSC)** pipeline, the V_{hin} sawtooth enters the circuit through the **BNC** connector and is labelled at the input node as seen in figure 3.21. The expected input range is -5 V to 5 V from the auxiliary output of the HP8552B **IF** section. A $100 \text{ k}\Omega$ resistance is connected between the non-inverting input (pin 3) of the LM358 op-amp (U1A) in order to set the input impedance and pull the voltage input voltage to 3.3 V . Additional $68 \text{ k}\Omega$ and $220 \text{ k}\Omega$ resistances contribute to the amplification as well as establishing a reference voltage at the non-inverting input.

The 0 V to 3.3 V conditioned output, V_{hout} is extracted through male connector pins as illustrated in figure 3.22.

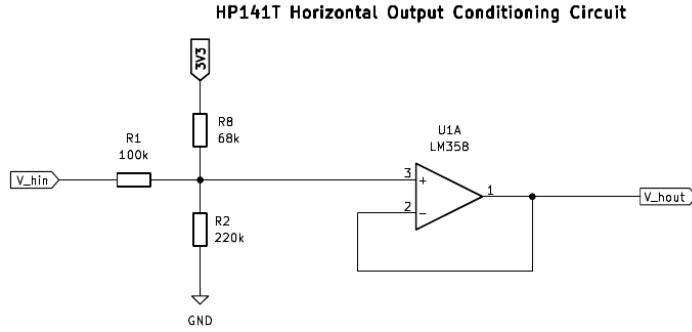


Figure 3.21: The [HSC](#) pipeline uses an LM358 op-amp to process the horizontal output signal from the HP141T scaling and shifting it to match the sampling microcontroller's [ADC](#)'s 0V to 3.3V input range.

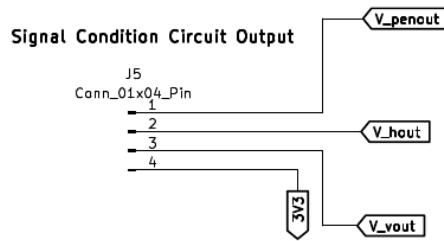


Figure 3.22: Circuit schematic of the connectors with 0V to 3.3V outputs corresponding to the digitized values of the vertical, horizontal, and pen-lift outputs of the [SCS](#).

The output of the [HSC](#) pipeline was simulated in LTSpice to determine the expected behaviour of the circuit. The simulation results are included in figure 3.23 where the output, in purple, is restricted to the desired input range of the microcontroller's [ADC](#). The simulation was run for 400 ms using the PWL command to reproduce the -5V to 5V input sawtooth as a piecewise function.

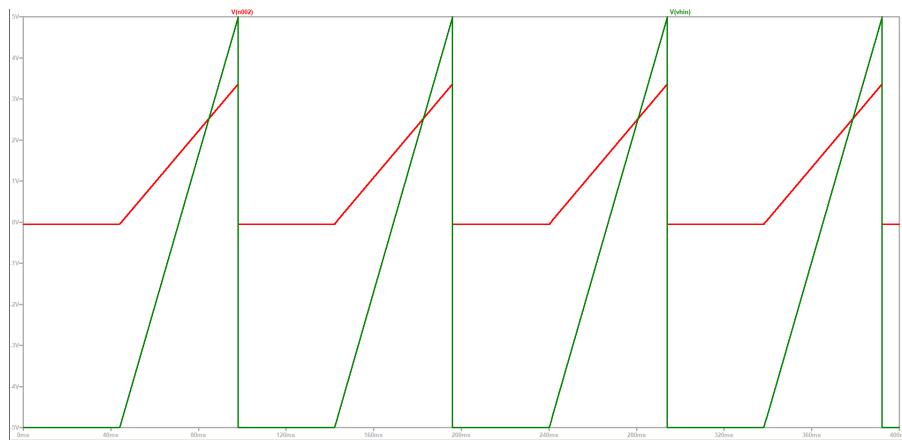


Figure 3.23: [HSC](#) simulation output showing that the op-amp configuration is expected to maintain the shape of the sawtooth scan output.

The specifications of the [SCS](#) with respect to the [HSC](#) pipeline are summarized in table ??.

Table 3.16: Input and output specifications of the [SCS](#).

The vertical output from the HP8552B is conditioned using LM358 op-amps in the inverting amplifier

configuration with a gain of $A_{vsc} = -3.7$ using a $10\text{ k}\Omega$ feedback resistor and a $2.7\text{ k}\Omega$ input resistor. This configuration was selected because the expected input voltage range of the [Vertical Signal Conditioning](#), V_{vin} , is -0.8 V to 0 V , and inverting and amplifying the small, negative voltages, is necessary for the operation of the [ADCs](#) on the STM32H723ZG. The [VSC](#) circuit is depicted in figure 3.24 with a $2.2\text{ k}\Omega$ compensating resistor at the non-inverting input of the LM358 op-amp to minimize the contribution of the input offset bias current.

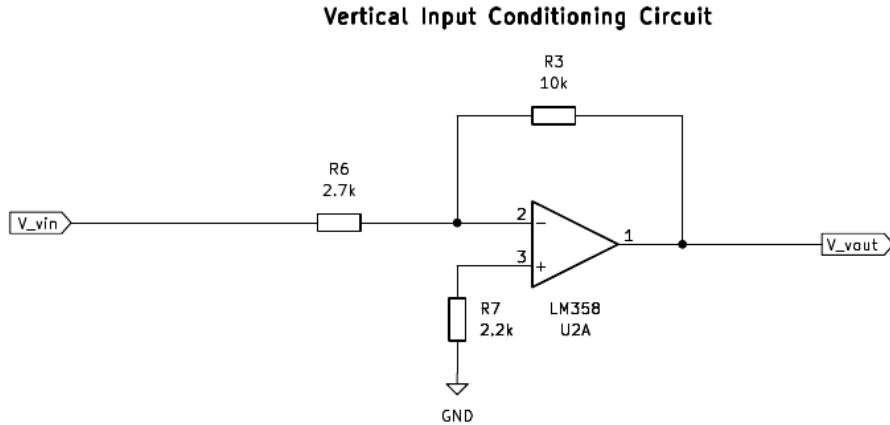


Figure 3.24: An inverting amplifier configuration is used in the [VSC](#) pipeline to prepare the vertical output of the HP8552B for digital processing.

The [VSC](#) circuit was simulated for different voltages and frequencies. A sinusoidal function was used in the simulation to model the fluctuations in the vertical output representing deflections in the amplitude of the input. To determine the behaviour of the [VSC](#) with respect to changes in voltage, a control frequency of 10 kHz was used. This frequency was selected corresponding to the calibration output of the HP8552B. Figure 3.25 shows the expected behaviour of the [VSC](#) circuit for a sinusoidal V_{vin} with a 0.8 V_{pp} amplitude. The output, V_{vout} , has a global minimum at 0 V and a global maximum at 3.0 V which is suitable for the input of the microcontroller's [ADCs](#). The plot shows that the shape of the input signal is maintained at the output, however, the output is out of phase by $\pi\text{ rad}$. In other words, the input is flipped and scaled by the inverting amplifier configuration.

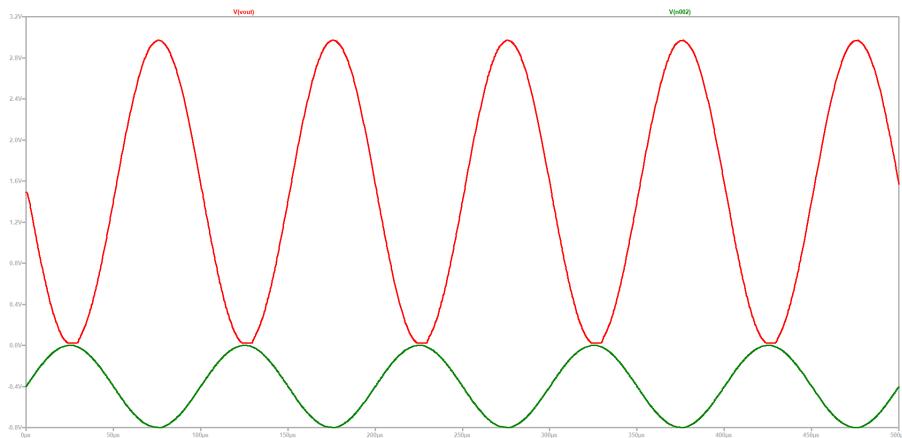


Figure 3.25: Simulation output for the case where V_{vin} assumes all values in the range between -0.8 V to 0 V .

Simulation results show slight distortions in the shape of the output for input voltages starting from -0.4 V as shown in figure ???. The distortions become more apparent for V_{vin} values that are lower than or equal -0.32 V as seen in figure 3.26 where the shape of the output is clipped at 0 V.

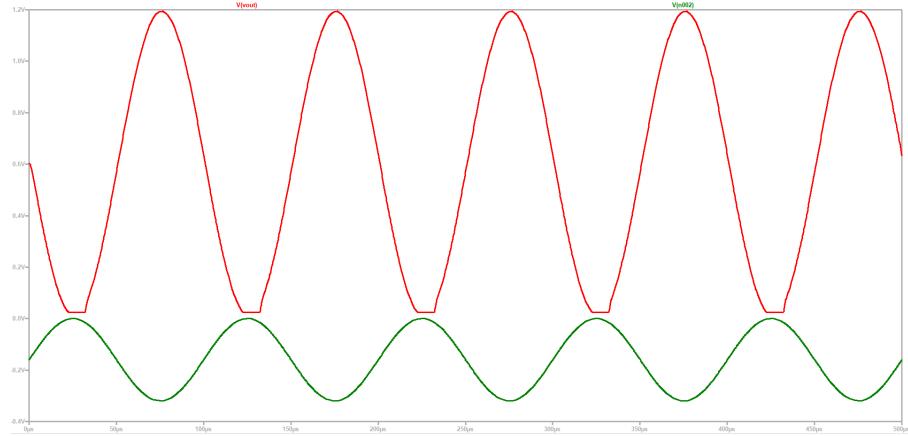


Figure 3.26: Simulation shows significant clipping of the output at 0 V when V_{vin} has a peak-to-peak voltage of -0.32 V.

Voltage clipping of the **VSC** pipeline output indicates a loss in the accuracy of the amplitude. According to the datasheet, the linear accuracy of the HP8552B in the vertical output is $0.1 \mu V$. However, simulations results showed that the sensitivity of the **VSC** circuit is $70 \mu V$, as seen in figure 3.27, corresponding to the voltage where visible change in the output start to be seen.

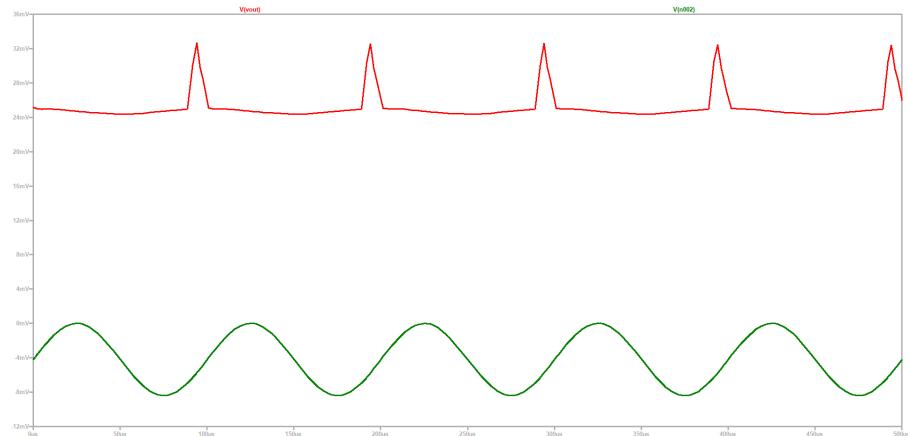


Figure 3.27: Simulation results indicate that the **VSC** circuit has a linear sensitivity of $70 \mu V$, compared to the linear sensitivity of $0.1 V$ in the vertical output of the HP8552B.

The above simulation results show that at 10 kHz, the **VSC** circuit significantly distorts the shape of the vertical output from the HP8552B **IF** section for voltages near 0 V. However, the performance of the circuit increases as the frequency increases. For example, no distortion or voltage clipping is evident for a sinusoidal input with a minimum value of -0.32 V when the vertical output from the HP8552B has a frequency of 100 kHz, as illustrated by figure 3.28.

At 300 kHz, which is the expected maximum frequency in the vertical output of the **IF** section, the shape of the output is not affected, however, the phase difference between the input and the output is

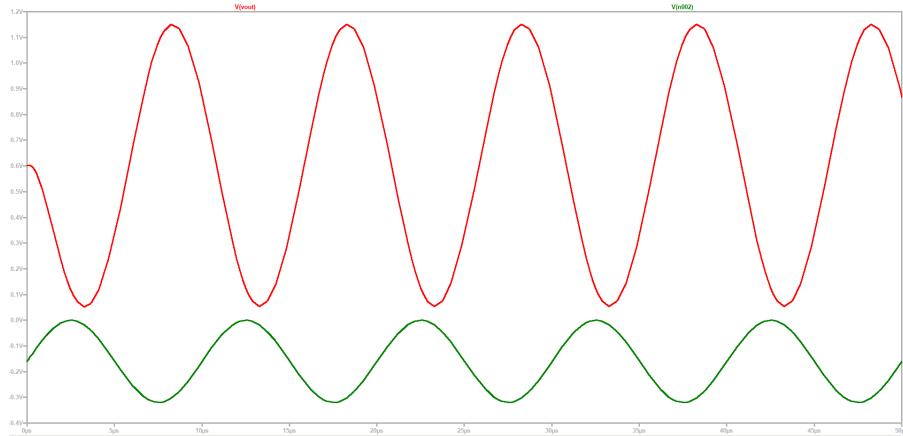


Figure 3.28: Showing reduced distortion and voltage clipping in the output for a vertical output frequency of 100 kHz.

affected as seen in figure 3.29. In addition, an offset of 37.27 mV is introduced at to the output of the **VSC** circuit at higher frequencies of V_{vin} .

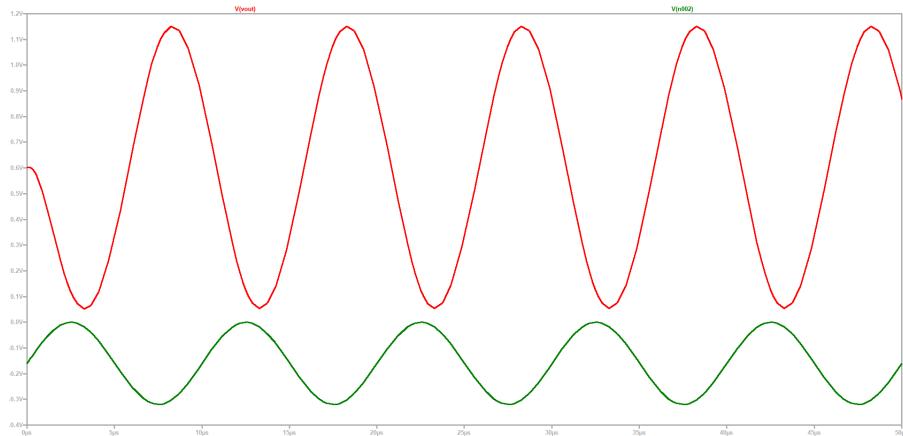


Figure 3.29: At the maximum rated vertical output frequency of 300 kHz, the phase difference between the input and the output of the **VSC** is slightly more than π rad.

The specification of the **SCS** with respect to the **VSC** pipeline are summarized in table ??.

Table 3.17: Input and output specifications of the **VSC**.

The circuit for the **Pen-Lift Signal Conditioning** pipeline is illustrated in figure 3.30 where an LM358 op-amp is used in the non-inverting buffer configuration which serves to scale the pen-lift output, V_{penin} , from the HP8552B plug-in of the HP141T system.

The expected input voltage range of the **PLSC** circuit is 0 V to 14 V, however, a LTSpice simulation of the circuit indicates that the maximum allowed value of V_{penin} is 18.77 V, corresponding to an output voltage of 3.3 V. Simulation results are shown in figure 3.31, in which a pulse function was used to represent the pen-up (18.77 V) and pen-down (0 V) states.

A full 3D model of the **SCS PCB** is depicted in figure 3.32a which uses the 8-pin chip with 2 LM358s in on packaging. Since 3 op-amps are used in the **SCS**, a total of 2 chips is required for the full

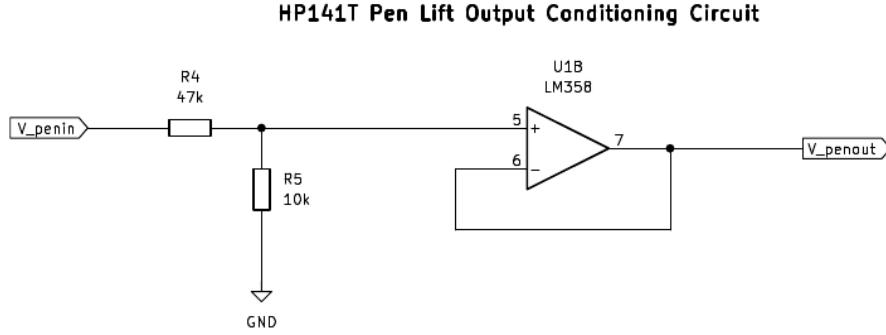


Figure 3.30: Schematic of the PLSC circuit for preparing the HP8552B pen-lift output for digital processing.

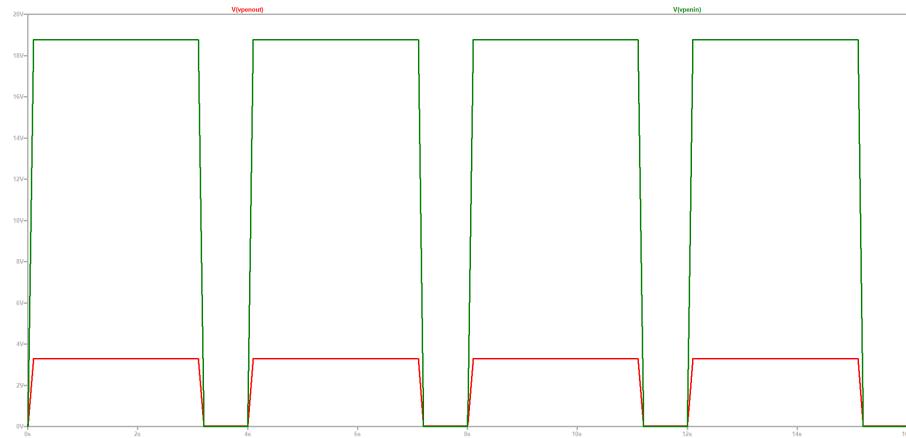
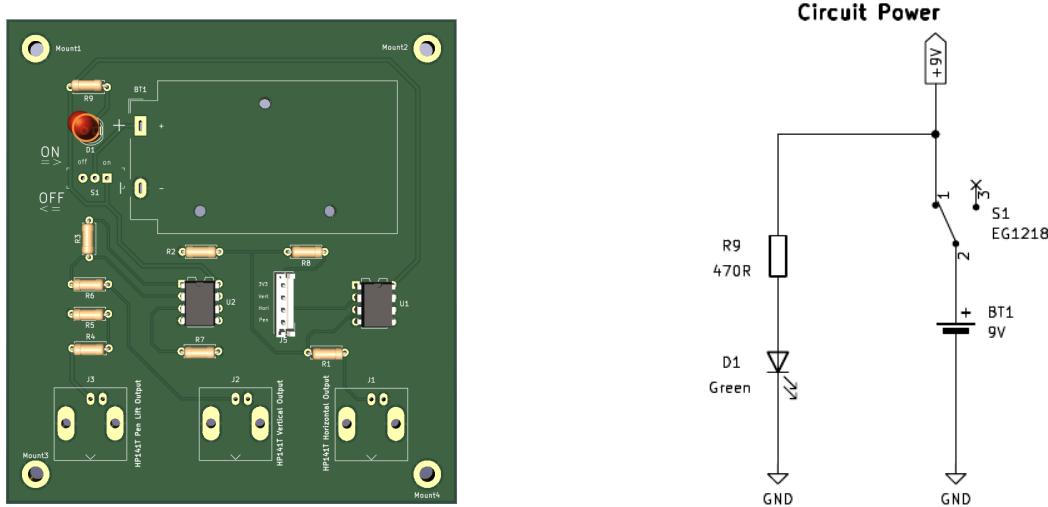


Figure 3.31: Simulation result corresponding to the maximum operation voltage of the PLSC circuit corresponding to 18.77 V.

implementation of the subsystem. Op-amps are powered by a 9 V battery as illustrated in figure 3.32b. Connection to the power source is controlled using a 3-pole switch. When the switch is ON, the power LED is active and when the switch is OFF, no power is supplied to the PCB and the LED is inactive.



(a) A 3D model of the SCS PCB.

(b) Schematic of the SCS power supply circuit.

Figure 3.32: The SCS is implemented on a single PCB with a single 9 V power source.

The physical hardware specifications of the **VSC PCB** are shown in figure ???. The **PCB** was designed using KiCad 6 to consist of a porous GND copper layer for establishing a reference voltage for the through-hole components. Surface mount holes are also grounded to prevent electric shock and ensure safety.

Table 3.18: Input and output specifications of the **VSC**.

Data Acquisition Subsystem

The **Data Acquisition Subsystem (DAS)** is responsible for converting conditioned analog signals from the **SCS** into digital values for further processing. Therefore, the **DAS** forms a critical part of the digitization of the HP141T system. It enables subsequent manipulation by the **Digital Processing Subsystem** and **GUIS**. The **DAS** ensures that the output analog signals from the **SCS**, i.e. V_{hout} , V_{vout} and V_{penout} , which are the scaled and shifted versions of the outputs from the HP8552B plug-in, are accurately sampled and converted into digital values suitable for processing by a **MCU** and **SBC**.

The diagram in figure 3.33 illustrates the functionality of the **DAS** where an **ADC** samples the analog inputs at a rate sufficient for capturing high frequency components (up to 300 kHz for the vertical channel). The digitized output values of the **DAS** represent the amplitude and timing of the signals which is crucial for reconstructing the frequency spectrum of the input signal at the input of the HP8555A **RF** section.

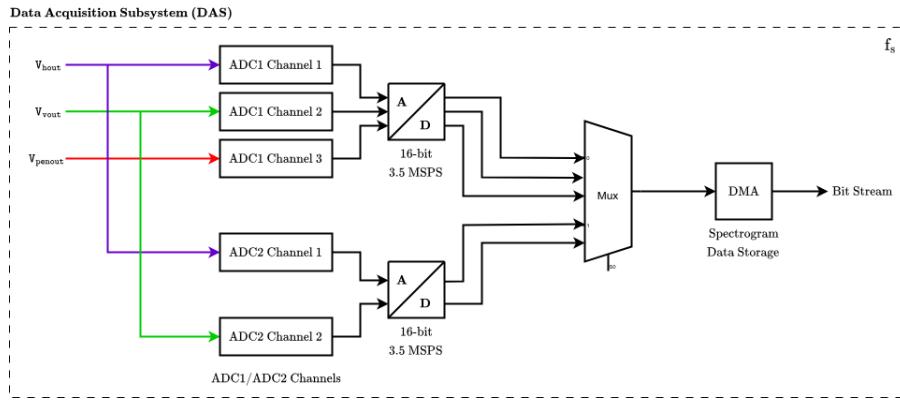


Figure 3.33: **DAS** subsystem block diagram showing the digitization of the analog signals using 2 16-bit **ADCs** operating in double-interleaved mode.

The **DAS** uses the two 16-bit resolution **ADCs** in double-interleaved mode. Given that the outputs of the **SCS** assume values up to a reference voltage of 3.3 V, the **DAS** can represent 65565 values in 50.35 μ V intervals. In 16-bit mode, the sampling rate of each **ADC** can reach up to 3.5 **MSPS** which implies that the system throughput can reach a total 7 **MSPS** when the **ADCs** are configured in dual fast interleaved mode as illustrated in figure 3.34.

In dual mode, ADC1 is the master and ADC2 is the slave. The pen-lift signal from the **SCS** behaves like a trigger which is internally synchronised for channel conversion. In dual fast interleaved mode, each **ADC** converts the channel every 14 **ADC** clock cycles and are stored into the ADC1 data register

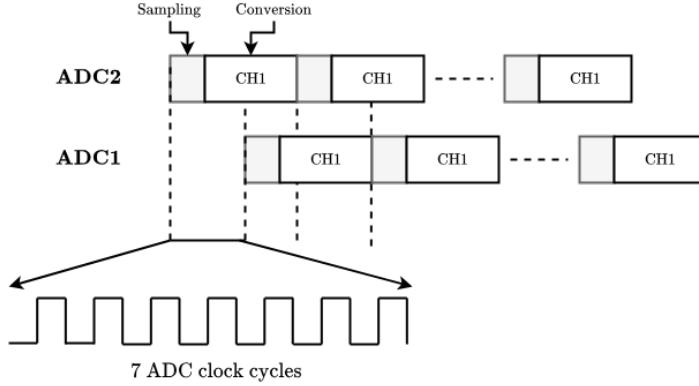


Figure 3.34: ADC1 and ADC2 convert the voltages in a period of 7 ADC clock cycles on the same channel.

in a 32-bit format [26]. The Nyquist criteria requires that the sampling frequency rate should be higher than or equal to twice the frequency of the signal to be converted. Since the maximum frequency of the signals from the SCS circuit is 300 kHz, a 3.5 MSPS rate meets the 600 kSPS needed for the vertical output. To ensure high integrity in data transfer, the Data Acquisition Subsystem (DAS) subsystem uses DMA2 for the master ADC and DMA1 for the slave.

Results from a MATLAB simulation of the dual interleaved ADC operation are shown in figure 3.35 where ADC1 and ADC2 alternate samples to double the sampling rate in order to satisfy the required 801 data points. The simulation calculates a SNR of 78 dB with 1 mV Gaussian noise. A table of summarizing the specifications modelled in the simulation is shown in ??.

Table 3.19: Specifications of the DAS.

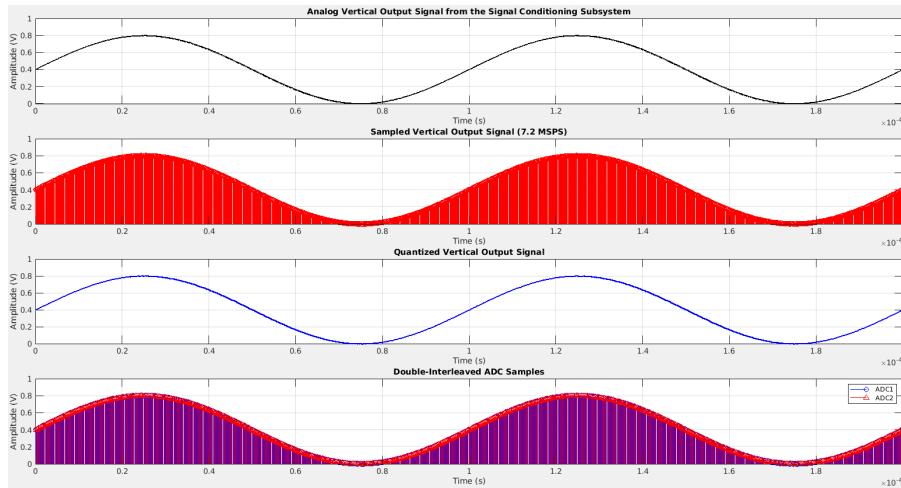


Figure 3.35: Simulation results of a well-behaved sinusoidal signal that is sampled by the ADCs on the STM32H723ZG development board.

Figure 3.35 shows the results from running the simulation with a well-behaved, sinusoidal analog vertical output signal from the SCS with a frequency of 10 kHz and a conditioned peak-to-peak voltage of 0.8 V. Given that the ADCs are operating with a 16-bit resolution, the subsequently high number of quantized

levels allow for a smooth sample with closely spaced data points as seen in the second and third plots in the figure. The final figure illustrates the output of the **ADC**s operating in double-interleaved mode. Samples from the master **ADC** are shown in red while samples from the slave are shown in blue.

Results from a more accurate simulation are shown in figure 3.36 where the **SCS** outputs a noisy signal as input to the **DAS**. The results illustrate that as long as the voltage is within the range 3.3 V, the double interleaved mode is able to capture a large number of samples to closely represent the input signal. The **DAS** propagates errors from the **SCS** circuit which prepares the analog outputs from the HP8552B.

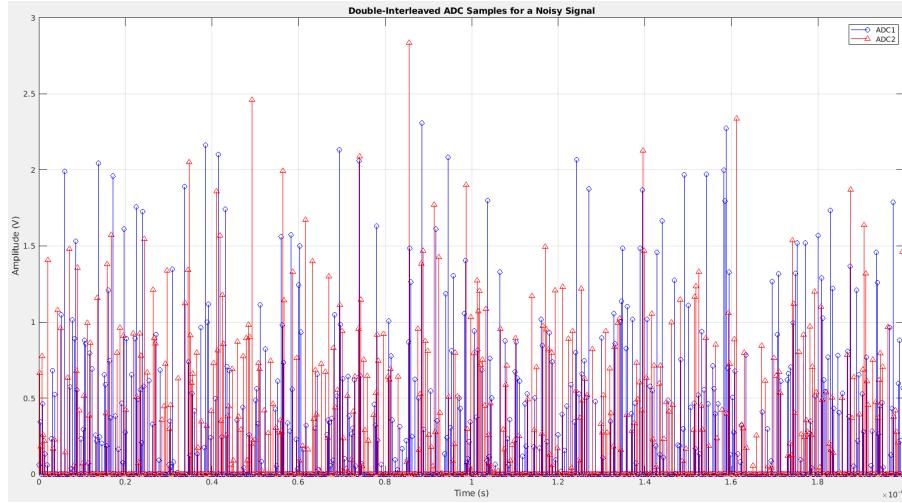


Figure 3.36: Simulation of a noisy signal sampled in double-interleaved mode by ADC1 and ADC2 on the STM32H723ZG.

The behaviour of the **DAS** is implemented on the STM32H723ZG **MCU** using embedded C to sample signals using functions in the **Hardware Abstraction Layer (HAL)** library. The horizontal (V_x), vertical (V_y) and pen-lift (V_z) outputs are mapped to PF10 (**ADC1_PIN8**), PF9 **ADC1_PIN7** and PF8 **ADC1_PIN6**, respectively. The code snippet in listing A.1 in the appendices shows the definition of the macros including the **DMA_REQUEST_ADC1** macro which enables the system to stream data to a buffer using **DMA**. The **DMA2_Stream0_IRQHandler** macro handles **DMA** interrupts for the **ADC** data streaming by calling the **HAL DMA** handler to process events such as a full buffer. The **DAS** uses a **uint16_t** buffer size of 802, corresponding to 401 samples per **ADC** channel.

The STM32H723ZG shares the same clock for the master and slave **ADC**s, derived from the peripheral clock PLCK2 to be divided by a factors of 2. The proposed design derives the **ADC**s' clock via a **ADC** prescaler (**ADC_CLOCKPRESCALER_PCLK_DIV4**) of 4 and a APB2 prescaler of 2, to achieve a sample achieve a frequency of 36 MHz. The sampling rate depends on the **ADC** clock frequency F_{ADC} , sampling period T_{ADC} , and conversion time T_{CONV} . At 36 MHz and a total of 19 cycles per sample, a single **ADC**'s sampling period is 0.528 μ s per sample (or 1.89 **MSPS**). The full sampling capability of the **ADC**s can be achieved by setting the **ADC** clock to 60 MHz and using 18 cycles per second, giving a sample rate of 6.66 **MSPS**. The **adc_error_flag** global variable is initialized to zero and is used to track **ADC** initialization errors for debugging.

The **HAL_DMA_IRQHandler** function from the STM32H7 **HAL API** is used to handle **DMA** interrupts for

the master **ADC** and triggers callbacks. Given that only the master's **DMA** handles all data transfers, no **DMA** is required for the slave **ADC**. Processed **ADC** data is stored using the **spectrogram_t** structure which is initialized as a pointer as shown in listing A.2.

```

1 void adc_error_handler(void) {
2     adc_error_flag = 1;
3     while(1) {}
4 }
5
6 spectrogram_t *adc_spectrogram = 0;
7
8 void adc_set_spectrogram(spectrogram_t *s) {
9     adc_spectrogram = s;
10}
11

```

Listing 3.1: The **HAL_DMA_IRQHandler** function is used to handle **DMA** transfers while the **adc_set_spectrogram(spectrogram_t *s)** function is used to initialize a pointer to a **spectrogram** structure.

The **adc_init** function forms a critical part of the **DAS** because it initializes ADC1 as the master and ADC2 as the slave in double-interleaved mode, configures three sampling channels, and begins **DMA** transfers as shown in listing A.2 in Appendix A. In addition, the function sets the resolution to 16-bits and enables a continuous mode for uninterrupted conversions for seamless real-time data acquisition. The development board is configured by the **ADC_DUALMODEDATAFORMAT_32_10_BITS** variable to use outputs with 32-bit words, however, the **adc_buffer** is **uint16_t**. Therefore, the **HAL** is expected to extract the lower 16 bits.

ADC data that is stored in the **adc_spectrogram** structure is processed using conversion callbacks such as the **HAL_ADC_ConvHalfCpltCallback** in listing 3.3 used when the **DMA** is half-full. The function reads interleaved data, corresponding to values of V_x , V_y and V_z from the **adc_buffer**. Three variables, **x_val**, **y_val** and **z_val** are scaled and inverted according to the 16-bit resolution of the **ADCs**. For example, **x_val** is scaled to **npoints** by dividing by 65536, corresponding to a 16-bit scaling which matches the resolution of the **ADCs**. The pen-lift output, V_z , is assigned a threshold value of 32768, corresponding to 1.2 V for the **pen-down** state as shown in the simulation results. The threshold value is expected to require adjustments if the **SCS** output is slightly higher or lower than the simulated outputs.

```

1 void HAL_ADC_ConvHalfCpltCallback(ADC_HandleTypeDef* AdcHandle) {
2     for (int j = 0; j < ADC_BUF_SIZE/6; ++j) {
3         uint32_t x_val = adc_buffer[j*3]; // V_x
4         uint32_t y_val = adc_buffer[j*3+1]; // V_y
5         uint32_t z_val = adc_buffer[j*3+2]; // V_z
6         x_val = (x_val * adc_spectrogram->npoints) / 65536; // 16-bit scaling
7         y_val = adc_spectrogram->size_y - (adc_spectrogram->size_y * y_val) / 65536;

```

```

8     z_val = z_val > 32768 ? 1 : 0; // Threshold at mid-scale (~1.65 V)
9     if (x_val < 5) prev_x = 0;
10    if (x_val > prev_x && z_val == 0) {
11        if (pending_normalization) {
12            adc_spectrogram->data_normal[x_val] = adc_spectrogram->size_y/2 - y_val;
13        }
14        adc_spectrogram->data[x_val] = y_val + adc_spectrogram->data_normal[x_val];
15        prev_x = x_val;
16    }
17 }
18 }
19

```

Listing 3.2: Showing conversion callbacks for processing ADC data when the DMA fills half.

The code design implements the `HAL_MspInit` callback function, shown in listing 3.3 to perform system level initializations as indicated in the `HAL` manual for the STM32H7 series. This function enables clocks for the ADCs, GPIOF, and DMA2, in addition to configuring the correct pins as analog inputs. Clocks and GPIO pins are enable with analog mode and no pull up resistors.

```

1 void HAL_ADC_MspInit(ADC_HandleTypeDef *hadc) {
2     GPIO_InitTypeDef GPIO_InitStruct = {0};
3     static DMA_HandleTypeDef hdma_adc;
4
5     if (hadc->Instance == ADCx_MASTER) {
6         /* Enable clocks */
7         ADCx_CLK_ENABLE();0
8         ADCx_CHANNEL_GPIO_CLK_ENABLE();
9         DMAx_CLK_ENABLE();
10        //...
11        /* Configure GPIO pins */
12        GPIO_InitStruct.Pin = ADCx_CHANNEL_PIN | ADCx_CHANNEL_PIN2 | ADCx_CHANNEL_PIN3;
13        GPIO_InitStruct.Mode = GPIO_MODE_ANALOG;
14        GPIO_InitStruct.Pull = GPIO_NOPULL;
15        //...
16        hdma_adc.Init.PeriphDataAlignment = DMA_PDATAALIGN_HALFWORD; // 16-bit
17        hdma_adc.Init.MemDataAlignment = DMA_MDATAALIGN_HALFWORD;
18        hdma_adc.Init.Mode = DMA_CIRCULAR;
19        //...
20        __HAL_LINKDMA(hadc, DMA_Handle, hdma_adc);
21
22        /* Configure NVIC */
23        HAL_NVIC_SetPriority(ADCx_DMA_IRQn, 0, 0);

```

```

24     HAL_NVIC_EnableIRQ(ADCx_DMA_IRQn);
25 }
26

```

Listing 3.3: Showing conversion callbacks for processing ADC data when the DMA fills half.

Note that the DMA is used in circular mode to support continuous data acquisition. The DMA_PDATAALIGN_HALFWORD variable aligns with 16-bit data corresponding to half a 32-bit word. Furthermore, the board is designed to use and Nested Vectored Interrupt Controller (NVIC) as a higher-priority interrupt for ensuring timely DMA handling for real-time data acquisition.

Due to time constraints and low availability of the STM32H723ZG development board, the device includes the output spectrogram of the DAS is stored in the .csv format with 4 columns for the time-stamp (ms) and the interleaved data samples corresponding to V_x , V_y , and V_z (in V).

HP141T system samples stored in the .csv file differ from HP141T Emulator samples based on the simulation results. For example, the simulated output at the STO pin of the XR2206 chip in the vertical output emulator circuit is expected to range between 0 V to 3.0 V instead of the 0 V to 3.3 V range output from the SCS circuit for inverting the original output from the HP141T. Note that although the sampled data is used to construct a spectrogram in the frequency domain, the actual values are interpreted in the time domain - directly associated with data points in the amplitude vs frequency plots representing the frequency domain characteristics of the inputs signal.

A Picoscope 2204A USB oscilloscope was selected as an alternative device for sampling because of its ability to acquire data at a maximum sampling rate of 100 MSPS, which would sufficiently satisfy requirements UR01 and SR01. The Picoscope 2204A is the smallest oscilloscope in the 2000 Series, which includes a range of 8-bit oscilloscopes with sample rates of up to 1 GSPS. In addition, the hand-held oscilloscope can output data in a group of .csv files with two columns for the time-stamp and sampled voltage, separately for two channels. Both alternative sampling devices (STM32H7 and Picoscope) aim to satisfy the design requirement for enabling trace storage capabilities (UR06).

Given that the Picoscope has two channels, A and B as shown in figure 3.37, data acquisition using the hand-held oscilloscope is performed continuously for the vertical and horizontal outputs. The pen-lift output can be analysed on an isolated channel during unit tests for ensuring that hardware subsystems are operating as expected. Alternatively, the vertical and pen-lift outputs can be used without the scan output by plotting interpolated samples of the vertical output. To sample data, the oscilloscope uses probes which can connect to connector pins on the HP141T Emulator and SCS circuits.

Picoscope samples can be visualized using the Picoscope 7 Software which includes multiple analysis tools such as visualizations of the amplitude vs frequency spectrum of the input. In addition to being a substitute sampling device for the STM32H723ZG board, the design employs the USB oscilloscope and accompanying Picoscope 7 Software to verify the operation of the HP141T Emulator and SCS circuits. The Picoscope 7 visualization tool display is only used during testing to determine if hardware operates as expected - in line with the behaviour of the simulated circuits.



Figure 3.37: Illustrating how the outputs of the HP141T can be sampled using input channels A and B on the Picoscope 2204A.

3.4.3 Digital Processing Subsystem

The aim of the **DPS** is to prepare the data for display in the **GUIS** and fulfil the requirement for the digitized **SA** to have 3 display modes associated with UR02. The subsystem uses Python to accomplish the tasks associated with each display mode. For example, the peak hold mode uses the `max()` function to find the maximum amplitude and corresponding frequency in the spectrogram data stored as an iterable object. Since spectrogram data can be retrieved from the STM32H7 **ADC** samples or from values sampled by the hand-held usb oscilloscope, construction of the spectrogram is considered as the first step in the flow of the **DAS**, illustrated in figure 3.38 which demonstrates how the time domain signal is processed to display the frequency domain characteristics of an input to the HP8555A plug-in. Most of the data processing occurs on the Raspberry Pi 4B, which is proposed for handling digital processes and displays through a **LCD** touch-screen in the **GUIS**.

The flow diagram shows that the methods used for processing sampled data depends on the whether input data is from the standard data from the HP141T system or the HP141T Emulator for cases where access the **SA** is limited. In other words, if the system being measured is the HP141T system with the original outputs, then the **Standard Voltage Scaling and Interpolation Schema (SVSIS)** is used to convert the data to appropriate values for the **Hardware Video Scaler (HVS)** which is later used by the **GUIS** for drawing the plot on the touchscreen. If measurements of the HP141T Emulator are taken, then the **Emulator Voltage Scaling and Interpolation Schema (EVSIS)** is used to scale the time domain values in the iterable spectrogram data. In the subsequent steps, the **DPS** assesses the user-selected display mode to determine which operation to perform. Different functions, such as the `average_spectrogram()` which updates the frequency bin's average at each scan. This subsection describes the scaling and interpolation algorithms, and the principle functions for each display mode used to process the data in the **DPS** for display in the **GUIS**.

To ensure that the **DAS** can integrate seamlessly with the **Digital Processing Subsystem (DPS)** and **Graphical User Interface Subsystem (GUIS)**, regardless of the sampling device that is available between the STM32H7 and Picoscope, the code for the spectrogram structure, adopted from a similar project designed to run on an STM32F746G Discovery board, was modified to process sampled data from 3

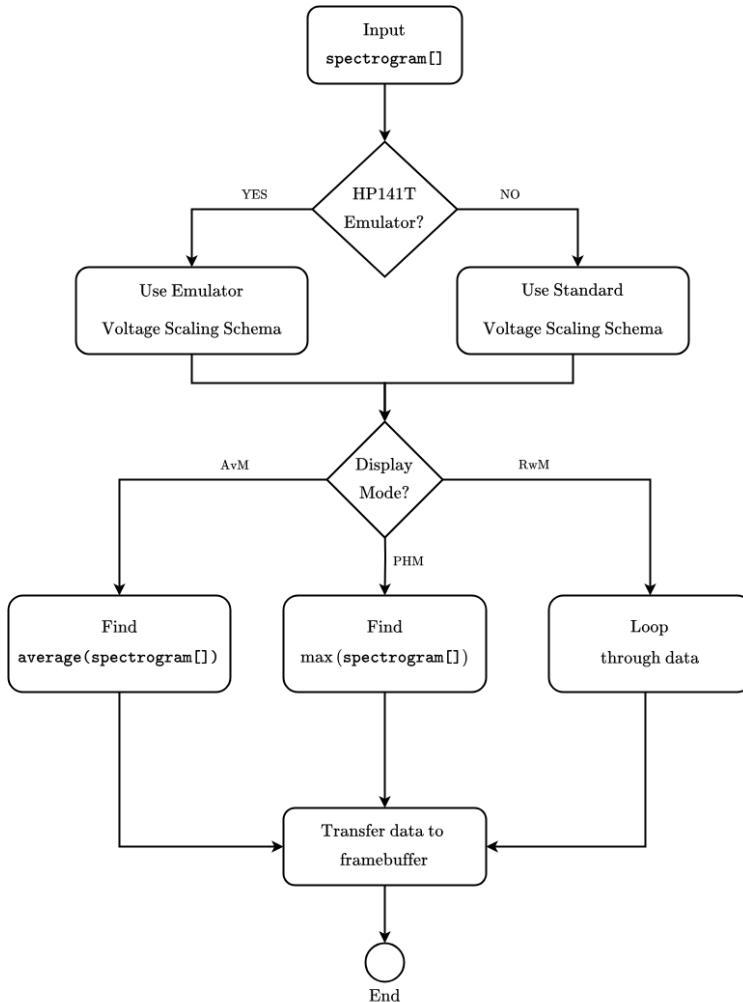


Figure 3.38: Flow diagram of the [Digital Processing Subsystem](#) which prepares time domain data for display in the frequency domain through the [Graphical User Interface Subsystem](#).

ADCs and store it in a series of .csv files, as shown in listing 3.4 [27].

```

1 int spectrogram_to_csv(spectrogram_t *s, const char *base_filename) {
2     if (!s || !base_filename) return -1;
3
4     const double sample_period_ms = 0.000138889; // 1 / 7.2 MSPS
5     int file_index = 0;
6     int point_count = 0;
7     char filename[64];
8     FILE *fp = NULL;
9
10    for (int i = 0; i < s->npoints && !pause_flag; ++i) {
11        if (point_count == 0) {
12            // Open new file
13            sprintf(filename, sizeof(filename), "%s%d.csv", base_filename, file_index);
14            fp = fopen(filename, "w");
15            if (!fp) return -1;
  
```

```

16     fprintf(fp, "Time (ms),V_x (V),V_y (V),V_z (V)\n");
17 }
18
19     double time_ms = i * sample_period_ms;
20     float vx = s->data_x[i] * 3.3f / 65535.0f;
21     float vy = s->data_y[i] * 3.3f / 65535.0f;
22     float vz = s->data_z[i] * 3.3f;
23
24     fprintf(fp, "%.8f,%.8f,%.8f,%.8f\n", time_ms, vx, vy, vz);
25
26     point_count++;
27     if (point_count >= 801 || i == s->npoints - 1) {
28         fclose(fp);
29         point_count = 0;
30         file_index++;
31     }
32 }
33
34     return 0;
35 }
36

```

Listing 3.4: Code snippet of the function for storing the interleaved sampling data in a .csv.

The `spectrogram_data_i.csv` ($i = 0, 1, 2, \dots, n$, for $n \in \mathbb{N}$) files generated by the `spectrogram_to_csv` function are inputs to the `DPS` where mathematical operations, such the max function for returning the largest value seem, are performed on the data in order to display it accurately and allow different view modes. To approximate real-time performance, a `pause_flag` variable set in the `pause_recording` function and triggered by a touch-screen button on the Raspberry Pi 4B `SBC`-based `GUI`. Time-stamps are calculated at intervals of 0.000 138 89 ms in order to fulfil high-sampling rate requirements. The function also scales the sampled 16-bit values to the appropriate 0 V to 3.3 V range.

A `Raspberry Pi (RPI)` 4B, as depicted in figure 3.39, was selected as the `SBC` for performing data processing algorithms in the `DPS`. The 4B is based on a ARM Cortex-A72 processor with a maximum CPU clock frequency of 1.5 GHz. In addition, the `RPI` has a versatile framebuffer which supports a resolution of up to 4K. The selection of the `RPI` 4B was also aimed at fulfilling the modernization requirement (UR01) which states that the display of the proposed design should behave like the display of newer `SAs`. The `RPI` is mounted to the touchscreen in the `GUIS` and is powered by a single wall wart power source, in alignment with requirement UR09.

The `RPI` 4B can also connect to both the STM32H723ZG microcontroller or Picoscope 2204A via `USB` to integrate the `DPS` with the `DAS`. Figure 3.39 of the `RPI`. A `Raspberry Pi OS` image is flashed to a `SD` card using `Raspberry Pi imager`, then the Python scripts for running the `DPS` and `GUIS` operations is loaded and executed on the device.

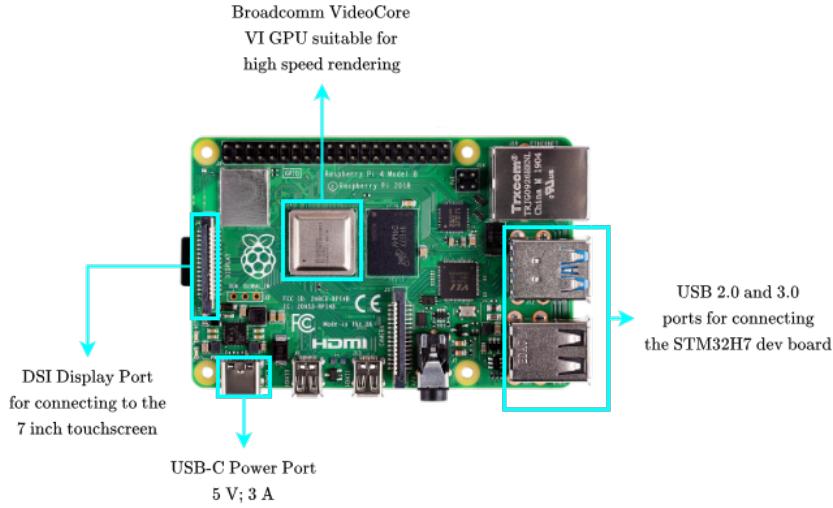


Figure 3.39: Parts of the RPI 4B that are of particular interest for this project.

Each division in the displayed amplitude vs frequency spectrogram is mapped to the sampled input voltages which are input to the DPS. Given that the vertical output of the HP141T is inverted and scaled in the SCS, the DPS associates the 0 V to 3.3 V v_y input with a vertical axis with values up to 80 dB . This is in line with the plot shown on the CRT display of the HP141T system which uses 0.1 V per division scaling to represent the amplitude on 8 divisions of 10 dB each (in line with UR03). Different stages of the vertical output are illustrated in figure 3.40 to inform the SVSIS and EVSIS for converting data to the appropriate scale and performing interpolation to ensure that a continuous spectrum can be displayed. The schematic of the flow of the vertical output at different stages in each

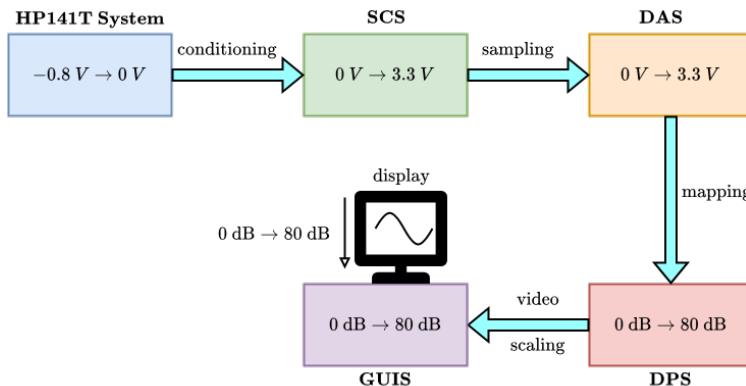


Figure 3.40: Illustrating the output voltage range associated with each subsystem. The operation performed by the subsystem is shown above the arrow preceding the block containing the output range of that subsystem.

subsystem shows that since the SCS inverts the scales the HP141T's auxiliary signal to 0 V to 3.3 V , the original negative maximum of -0.8 V is mapped to 0 V at the ADC stage. Similarly, the original zero minimum voltage maps to the maximum voltage of the ADC. To prepare the data for display on the amplitude vs frequency plot, the DPS maps the ADC's 0 V to 0 dB and uses a linear mapping to plot the points along the vertical axis. For example, given that 0 V to -0.8 V maps to 3.3 V to 0 V , the

ADC sample v_y can be converted to the 80-dB using

$$\text{dB} = 80 \times \frac{(3.3 \text{ V} - v_y)}{3.3 \text{ V}} \quad (3.4)$$

so that each 0.1 V shift corresponds to 10 dB.

Preparations for displaying the HP141T Emulator vertical output are performed in a similar manner to the one described above for scaling the vertical output of the HP141T **SA** to the required 80 dB amplitude range. However, the expected input range is mapped from [0 V, 3.0 V] to [80 dB].

The horizontal output is mapped to the frequency axis to represent the scan rate. The original output range of the HP141T system's sawtooth function is [-5 V, 5 V]. After the signal is conditioned through the operational amplifier-based circuit in **SCS**, values range between 0 V and 3.3 V. Time domain characteristics of the scan output do not change during the signal conditioning process - suggested by the simulation results. This implies that the time column in the .csv file is used directly to display the frequency axis on the display with 10 horizontal divisions. The sawtooth rise-time represents the frequency sweep. For 801 points at 7.2 **MSPS**, each division corresponds to 0.0111 ms.

The **DPS** assumes that the HP141T is operating in FULL SCAN mode where the whole tuning section band is displayed. The tuning section, or **RF** section, measures frequencies between 20 Hz and 18 GHz with the center frequency marked on the **CRT** display. The design is expected to work for the PER DIVISION mode which allows for different scan rates to enable signal analysis.

The **DPS** uses default linear interpolation associated with the `ax.plot` which is considered to be sufficient for this application given the large sampled data set. This is to say that the time between points (0.139 μ s) is small enough that linear segments should appear smooth in the **GUIS** display. The design explores alternative methods such as Spline interpolation which fits a piecewise polynomial through the data points. Implementation of Spline interpolation is expected to reduce sharp transitions that typically appear with linear interpolation by increasing the resolution to 1000 points so that the curve appears continuous. This type of interpolation is executed using the `scipy.interpolate` module's `CubicSpline` function which interpolate vertical output data in both the **SVSIS** and **EVSIS** as shown in the code snippet in listing A.3 in the appendix.

3.4.4 Graphical User Interface Subsystem

The **GUIS** uses **GUI** libraries such as `matplotlib` and `tkinter` to render the amplitude vs frequency plot on the digitized **SA**. The **GUI** libraries interface with the **RPI**'s framebuffer, through which the **DPS** sends video data for drawing directly to the screen. The final image frame is pushed directly to `/dev/fb0` to ensure that frequency domain data is rendered in real-time on the display based on the processed outputs of the **DPS**.

Table ?? gives a description of the properties of the **GUIS** which replicates the HP141T's **CRT** graticule.

The **GUIS** employs the `hp141t` Python module containing functions for enabling the **PHM**, **AvM**, and **RwM**. The module functions belong in the **DPS** for processing vertical output data and provides

Property	Description	Associated Values
Vertical Axis	Mapping of ADC samples to match the scaling on the original HP141T display.	80 dB scale (10 dB per division)
Horizontal Axis	Mapping of scan time to the frequency in alignment with the HP141T's sweep display	10 divisions of the scan outputs progression
Grid	A grid replicating the CRT graticule.	8x10 grid (8 amplitude divisions, 10 scan time divisions)
CRT Appearance	A black background colour with a green trace and grey grid lines to resemble the aesthetic of the CRT .	black = #000000; green = #0080000; grey = #808080
Touchscreen	TFT DSI capacitive touchscreen display which enables the user to access different modes.	7 inches with 800x480 resolution

Table 3.20: Highlighted properties of the **GUIS**.

interpolated outputs to the **GUIS**. Code snippets for the functions in the `hp141t` module are included in listing A.5 in the +appendices section.

The code for the **PHM** tracks and displays the maximum value seen for each time point and updated per scan when user selects the **PHM** button. For the **AvM**, the **DPS** computes a running average, updated at each scan, and the **GUIS** displays the output to the user when the **AvM** button is selected on the touchscreen. The **GUIS** also displays the latest value, overwriting previous values during each scan. During system initialization, the **GUIS** sets the default mode to **RwM** and initializes the `peak_data` and `raw_data` data arrays for storing the output associated with each mode. The output of the **AvM** is weighted by scan count.

Button functions set the modes independently so that the **GUIS** operates in one mode at a time. Each button function calls `processor.set_mode` and updates the display to ensure that the user maintains control. Buttons have a simple design with rounded edges, while adhering to Nielsen's design heuristics.

To fulfil the requirement for system to have tutorials and documentation to help users (SR06), a `show_help` function, shown in listing A.6, is associated with a **HELP** button which creates a pop-up window with a tutorial. The tutorial is displayed in a 400x300 pixel window with a black background. The content of the tutorial covers the purpose of the design, available modes, usage instructions and notes on the HP141T system.

3.5 Experimental Test Procedures

The following section details the methodology and setup used to verify the functionality and performance of the proposed design for the modernized HP141T **SA**. Experimental procedures are divided into three main categories, namely, unit tests, integration tests and acceptance tests. Unit tests are designed to validate the correctness of the individual subsystem software and hardware interfaces in isolation.

Integration tests are designed to evaluate the interactions between subsystems to ensure that the

subsystems work as expected with the HP141T system which includes the **IF** and **RF** sections. Acceptance tests assess the system's compliance with the user and system requirements shown in tables 1.1 and 1.2. Included are the test environments, tools and tasks for systematically evaluating the behaviour of each subsystem and the overall system. Tools for supplying power and performing measurements during hardware-based tests included:

- Siglent SDG1010 10 MHz function generator for producing the modelling the auxiliary outputs of the HP141T, as well as noisy signals for non-ideal cases.
- HPE3630A Triple Output DC Power Supply with 0 V – 6 V, 2.5 A outputs, or 0 V - \pm 20 V, 0 V - 0.5 A outputs.
- Picoscope 2204A **USB** oscilloscope with two analog channels and maximum input frequency of 10 MHz.
- Oscilloscope probes with 10x scaling capabilities.

The Picoscope 2204A was used together with the Picoscope 7 Software to measure analog signals in all experiments. The oscilloscope was connected to a **USB** 3.0 port of a Linux-based Dell Inspiron 15 machine and outputs were analyzed using the Picoscope 7 software on the machine.

3.5.1 Unit Tests for Subsystem Verification

HP141T System Unit Tests

Unit tests for the HP141T system were advised by the datasheets of the HP8555A and HP8552B plug-in sections. The following unit test procedures were considered for the HP141T system. In the description of each unit test, it is assumed that the Picoscope is connected via **USB** to the computer on which the Picoscope 7 software is executed.

UTA00: Vertical Output Voltage Range

Connect an input to the HP8555A with a frequency between 20 MHz to 18 GHz. Check that the Picoscope environment is running. Connect the vertical output (Y) of the HP8552B to channel A of the Picoscope 2204A using oscilloscope probes. Set the scaling to 10x on the oscilloscope and the probe connected to channel A. Ensure that the Vertical Channel Options for A are set to **Auto**, the voltage scaling to \pm 1 V, and that **Invert** is **Off**. The measured voltage on channel A should be between 0 V and -0.8 V. If there is no input at the HP8555A plug-in, then the voltage should be 0 V.

UTA01: Vertical Output Frequency Range

With the vertical output connected to channel A, and the Picoscope 7 software running, click on **View** and select **spectrum**. The spectrum should contain frequency components between 10 Hz and 300 Hz.

UTA02: Horizontal Output Voltage Range

Connect the horizontal output of the HP141T to channel B of the Picoscope and select the channel in the software. The signal should look like a sawtooth function ranging between \pm 5 V and a rise time of up to 1 div/ μ s.

UTA03: Pen-Lift Output Voltage Range

Disconnect the probes from HP141T [SA](#). Remove the probe connected to channel B. Connect the probe associated with channel A to the pen-lift output of the HP141T. The signal displayed in the oscilloscope's software should range between 0 V and 14 V.

HP141T Emulator Unit Tests

Unit tests for the HP141T Emulator were developed based on the expected behavior of the analog outputs and the digital processing pipeline. These tests verify the signal characteristics produced by the emulator and ensure that the ADC sampling, signal conditioning, and waveform generation circuits function as designed. It is assumed that a Picoscope 2204A is connected to the system and controlled via the Picoscope 7 software.

UTB00: Emulated Vertical Output Voltage Range

Run the emulator and ensure a test signal (sinusoidal or modulated RF) is applied to the ADC input. Connect the emulator's vertical output to channel A of the Picoscope. Ensure the vertical scaling is set to ± 5 V and the probe attenuation is 10x. The observed voltage should range between 0 V and 3.0 V.

UTB01: Emulated Vertical Output Frequency Range

With the vertical output still connected to channel A, switch to spectrum view in Picoscope 7. The frequency components of the vertical signal should be 10 Hz, consistent with the design.

UTB02: Emulated Horizontal Output Voltage Range

Connect the horizontal output of the emulator to channel B. The signal should appear as a ramp or sawtooth waveform with a voltage swing between -4.76 V and $+4.75$ V. The period should correspond to the configured scan time base (e.g., 1 ms to 100 s per sweep).

UTB03: Pen-Lift Output Voltage Range

Connect channel A to the emulator's pen-lift output. When scanning is active, the pen-lift output should be 0 V. During retrace (i.e., blanking interval), the output should rise to approximately 14 V. Confirm that this toggling corresponds with the sawtooth waveform transitions on the horizontal output.

UTB04: HP141T Power Supply

Ensure the HP141T Emulator is powered by the designated plug-in AC/DC adapter rated at 12 V, 1 A maximum output. With the emulator powered on, use a multimeter or oscilloscope to measure the voltage at the outputs of all power rails generated by internal regulators and inverter circuits. Confirm the following:

- **+3.3V rail:** Output must be within 3.3 V ± 0.1 V, used for ADCs and digital logic.
- **-5V rail (if generated):** Output must be within -5 V ± 0.2 V, used for op-amps or analog reference circuits.
- **+5V or +12V rail:** Check regulator outputs for subsystems such as op-amps or display drivers.

Use an oscilloscope to check for voltage ripple and noise on each power rail. Verify that ripple does not exceed 50 mV under load. Confirm that startup transients settle within 10 ms. This test ensures the emulator's power delivery is stable and within tolerance for all connected subsystems.

Signal Conditioning Subsystem Unit Tests

The **Signal Conditioning Subsystem (SCS)** is responsible for scaling, shifting, and inverting the auxiliary analog outputs from the HP141T system so they can be sampled accurately by the **ADCs**. Unit testing of this subsystem focuses on validating voltage range translation, signal polarity, and waveform integrity. All tests assume that the outputs are probed using a calibrated Picoscope 2204A with appropriate probe attenuation (10x) and configured channel settings.

UTC01: Vertical Output Inversion and Scaling

Power the **SCS PCB** by a 9 V DC supply - the green LED on the board should turn on to indicate that the circuit is operational.

UTC01: Vertical Output Inversion and Scaling

Apply a known signal to the vertical output of the HP141T (e.g., sinusoidal waveform with a peak-to-peak voltage between up to 0.8 V). Connect this output to the **SCS** input and probe the output with channel A of the Picoscope. Verify that the signal is inverted and scaled such that -0.8 V corresponds to 0 V and 0 V corresponds to approximately 3.3 V. Use the measurement cursor to confirm linear scaling.

UTC02: Horizontal Output Linear Scaling

Input a test sawtooth waveform into the horizontal output path of the HP141T. Connect the SCS output to channel B and compare the measured waveform to the known input. Confirm that the SCS output exhibits a full-scale ramp from -5 V to +5 V mapped linearly to the 0 V to 3.3 V range, preserving waveform shape and ramp symmetry.

UTC03: Pen-Lift Output Logic Level Shifting

Apply a manual scan trigger to the HP141T system to induce a pen-lift event. Connect the **SCS** output for the pen-lift signal to channel A. The output should transition from approximately 0 V (during scan) to a logic high voltage of 3.3 V when blanking is active. Verify that the logic levels remain within acceptable digital input thresholds for the **ADC** and digital processing stages.

UTC04: Signal Integrity and Noise Margin

With the **SCS** outputs connected to the Picoscope, enable high-resolution mode and apply bandwidth filtering to assess signal noise and distortion. Verify that output signals maintain consistent voltage levels without oscillation, excessive ripple, or overshoot. Measure peak-to-peak noise and confirm that the noise margin is within acceptable limits (e.g., less than 20 mV for a steady signal).

UTC05: Response to Overvoltage and Fault Conditions

Apply an out-of-range voltage (e.g., ± 10 V) to each input line of the **SCS** one at a time. Monitor the output with the Picoscope. Confirm that the subsystem clamps or safely rejects the signal without

passing excessive voltage to the **ADC** inputs.

Data Acquisition Subsystem Unit Tests

The **DAS** is responsible for sampling the conditioned analog signals from the HP141T Emulator, digitizing them using high-speed ADCs, and storing the values for further processing in the **DPS**. The following unit tests verify correct sampling behavior, signal integrity, timing, and storage functionality.

UTD00: ADC Voltage Input Range Test

Apply known **DC** voltages within the expected input range (e.g., 0V, 1.65V, and 3.3V) to each **ADC** input channel (V_x , V_y , V_z). Sample the inputs using the **DAS** and export the resulting values to a **.csv** file. Verify that the digital values correspond to the expected 16-bit quantization of the input voltages, using:

$$V_{in} = \frac{\text{ADC Output}}{65535} \times 3.3 \text{ V}$$

UTD01: Sampling Rate Verification

Configure the DAS to run at the target sampling rate (e.g., 7.2 **MSPS**). Using a signal generator, apply a known frequency signal (e.g., 1 MHz sine wave) to an **ADC** channel. Use the time stamps in the **.csv** output to compute the effective sampling interval and verify that the measured sampling rate is within $\pm 5\%$ of the configured value.

UTD02: Multi-Channel Synchronization Test

Apply identical signals to all three **ADC** input channels and run the sampling routine. Compare the sampled waveforms from the **.csv** output to confirm that the channels are sampled in sync or with known interleaving, based on the expected **ADC** configuration. Check for time alignment by comparing signal edges or zero crossings.

UTD03: Signal Integrity

Apply a low-noise sine wave signal to an **ADC** input and record the sampled data. Analyze the resulting waveform for quantization noise and distortion using FFT. Verify that the signal-to-noise ratio (**SNR**) and effective number of bits (**ENOB**) match the **ADC**'s datasheet specification within tolerance.

UTD04: File Writing

Run the `spectrogram_to_csv()` function after a complete data capture session. Open the generated **.csv** files and verify the format:

- Header row includes <Time (ms), V_x (V), V_y (V), V_z (V)>
- Each row is correctly formatted with numerical values
- All expected sample points are recorded (e.g., 801 samples per file)

UTD05: Pause/Resume Functionality

Use the touch interface to trigger a pause during data capture. Confirm that data acquisition halts immediately and resumes only when the pause flag is cleared. Ensure that partial files are properly

closed and new files are opened after resuming.

Digital Processing Subsystem Unit Tests

The **DPS** is responsible for interpreting sampled **ADC** data, applying transformations such as scaling and interpolation, and preparing the data for display in various user-selectable modes. The following unit tests verify the correctness of mathematical processing, mode switching, and data integrity.

UTE00: Spectrogram Data Scaling Verification

Load a **.csv** file with known voltage values into the **DPS** and process it using the standard vertical scaling equation:

$$\text{dB} = 80 \times \frac{(3.3 - V_y)}{3.3}$$

Verify that the resulting amplitude values match expected decibel levels (e.g., a 0V input maps to 80 dB and 3.3V to 0 dB).

UTE01: Peak Hold Mode Accuracy

Feed multiple spectrograms with varying amplitudes into the **DPS**. Confirm that the **max()** function correctly retains the highest amplitude observed for each frequency bin across scans.

UTE02: Average Mode Calculation Accuracy

Run the **average_spectrogram()** function over multiple scans. Confirm that each bin's output is the correct arithmetic mean of its values from all prior scans.

UTE03: Interpolation Output Continuity

Input coarse or gapped spectrogram data into the **DPS**. Verify that the interpolation algorithm produces a continuous, visually smooth spectrum with no missing or duplicated points.

UTE04: Spectrogram Data Pipeline Integrity

Verify that all expected data points from the **.csv** input are preserved throughout the DPS processing pipeline and correctly passed to the GUI plotting module.

GUIS Unit Tests

The **GUIS** runs on the Raspberry Pi 4B and is responsible for rendering the processed spectrogram data, managing user inputs via touchscreen, and switching between display modes. The following unit tests ensure correct visualization, responsiveness, and display scaling.

UTE05: Display Scaling Verification

Load a set of scaled decibel values from the **DPS** into the **GUI** rendering module. Confirm that each 10 dB increment correctly maps to one vertical division on the **GUI** plot.

UTE06: Touchscreen Mode Switching

Verify that all **GUI** buttons (e.g., Average, Peak Hold, Raw) correctly update the active display mode and the interface reflects the change without delay.

UTE07: Real-Time Plot Update Rate

Run the [GUI](#) using a constant stream of spectrogram data and measure the update rate. Verify that the display refreshes in near real-time (e.g., every 50–100 ms) with minimal lag.

UTE08: Graph Rendering Accuracy

Input a known waveform or fixed dataset into the [GUIS](#). Visually verify that the plot matches expected amplitude vs frequency behavior. Compare screen coordinates to underlying data values to confirm accuracy.

UTE09: Pause and Resume Functionality

Press the pause/resume button on the [GUI](#) while data is being processed. Confirm that data acquisition halts immediately, and resumes without loss of continuity or data corruption when resumed.

3.5.2 Integration Tests for Overall Digitized HP141T System Verification

Integration tests were performed to verify that the individual subsystems—namely the [SCS](#), [DAS](#), [DPS](#), and [GUIS](#)—function correctly when combined. These tests ensured that signals flowed correctly between stages and that each interface (analog, digital, and software) operated as expected. For example, outputs from the [SCS](#) were validated at the [DAS](#) input, and the data exported from the [DAS](#) was processed and visualized through the [DPS](#) and [GUIS](#). The integration tests focused on detecting mismatches in voltage ranges, data formats, timing synchronization, and communication between hardware and software components.

3.5.3 Acceptance Tests

Acceptance testing evaluated the system against the functional requirements defined early in the project, such as accurate signal acquisition, support for multiple display modes, and successful user interaction through the [GUI](#). These tests confirmed the system’s readiness for real-world use and assessed overall usability, responsiveness, and fidelity of the emulator in replicating the HP141T’s behavior. Each requirement was validated by comparing the digitized system’s output to reference measurements from the original HP141T system or expected theoretical behavior, ensuring that the design met its intended performance and user expectations.

Chapter 4

Results and Discussion

This chapter presents the results obtained from experimental testing of the digitized HP141T system and discusses their significance in the context of the system's performance, fidelity, and ability to meet the specified functional requirements. Each section corresponds to a key subsystem or system-level outcome, integrating quantitative measurements with qualitative assessments to evaluate accuracy, responsiveness, and usability.

4.1 HP141T Emulator Subsystem Test Results

This section presents the results of unit testing the HP141T Emulator [PCB](#), which is responsible for signal emulation and waveform generation. The tests focus on validating the behavior of the analog outputs (vertical, horizontal, and pen-lift signals) against expected electrical characteristics derived from the original HP141T system. Each test verifies waveform amplitude, frequency range, and functional timing to ensure compatibility with downstream processing.

The vertical output was found to swing linearly between 0 V and 3.3 V in response to a simulated input corresponding to the HP8552B's -0.8 V to 0 V range. The horizontal output produced a sawtooth waveform spanning approximately -5 V to 5 V , with adjustable scan rates verified between 1 ms and 100 s per sweep, matching the original system's 1-2-5 sequence. The pen-lift signal correctly toggled between 0 V during scan and 14 V during retrace, consistent with the CRT blanking requirement.

These results confirm that the HP141T Emulator [PCB](#) replicates the critical analog behaviors of the original system and can serve as a reliable signal source for downstream subsystems in the digitized HP141T platform.

4.2 Signal Conditioning Subsystem Results

This section presents the measured voltage transformations and waveform fidelity at the output of the [SCS](#). It compares the observed behavior against theoretical expectations based on the known input-output mappings, such as the inversion of the vertical output and scaling to the [ADC](#)-compatible voltage range. Signal integrity and noise performance are also discussed.

4.3 Data Acquisition Subsystem Results

The [DAS](#) results focus on the performance of the [ADCs](#), including sampling accuracy, effective resolution, synchronization across channels, and real-time data logging. This section evaluates how reliably the [DAS](#) converts conditioned analog signals into digital format and stores them in the required [.csv](#) structure for downstream processing.

4.4 Digital Processing Subsystem Results

This section presents the output of the [DPS](#), including processed amplitude vs frequency data derived from the sampled input. The effectiveness of the implemented algorithms for scaling, interpolation, peak detection, and averaging is assessed using visual plots and numerical comparisons with expected values.

4.5 Graphical User Interface Results

This section evaluates the performance of the [GUIs](#) in terms of visual accuracy, responsiveness, and user experience. Test results include display scaling verification, update latency, and the reliability of interactive controls for switching between display modes. Screenshots and user feedback may be used to support the discussion.

4.6 Integrated System Performance

Here, the combined behavior of all subsystems is evaluated to assess how well the full digitized HP141T meets its intended purpose. System level outputs are compared with those of the original HP141T to discuss emulation fidelity. Particular focus is placed on real-time performance, mode switching, and overall usability.

Chapter 5

Conclusions

The same rule holds for us now, of course: we choose our next world through what we learn in this one. Learn nothing, and the next world is the same as this one.

—Richard Bach, *Jonathan Livingston Seagull*

The purpose of this project was to...

This report began with...

The literature review was followed in Chapter...

The bulk of the work for this project followed next, in Chapter...

In Chapter...

Finally, Chapter... attempted to...

In summary, the project achieved the goals that were set out, by designing and demonstrating...

Chapter 6

Recommendations

It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced.

—Abraham Lincoln

6.1 Hardware Recommendations

To improve high current demand when sampling using the [ADCs](#) on the STM32H746ZG development board, a capacitor should be placed close to the [ADCs](#)' input pins. According to ST Electronics, the capacitor stores charge to ensure good settling time to the desired input voltage level.

Bibliography

- [1] D. Rytting, “Arftg 50 year network analyzer history,” in *2008 71st ARFTG Microwave Measurement Conference*. IEEE, 2008, pp. 1–8.
- [2] HP. (1972) Signal analyzers. [Online]. Available: <https://www.mikrocontroller.net/attachment/255249/HP141T.pdf#:~:text=The%20141T%20based%20spectrum%20analyzer%20features%20absolute,range%2C%20and%20simple%20to%20interpret%20display%20output.&text=Signals%20at%20as%20low%20a%20level%20as,the%20spectrum%20analyzer%20with%2010%20Hz%20bandwidth>.
- [3] W. Wolf and H. Deubel, “P31 Phosphor Persistence at Photopic Mean Luminance Level.” *Spatial Vision*, no. 4, pp. 323–333, 1997.
- [4] J. M. D. Pereira, “The History and Technology of Oscilloscopes,” *IEEE Instrumentation & Measurement Magazine*, vol. 9, no. 6, pp. 27–35, 2006.
- [5] D. Herres and D. Herres, “Waveforms and Instrumentation Overview,” *Oscilloscopes: A Manual for Students, Engineers, and Scientists*, pp. 1–24, 2020.
- [6] C. Völlinger and M. Wendt, “Rf Measurement Techniques,” pp. 1–51, 2023.
- [7] J. Deery, “The Real History of Real-Time Spectrum Analyzers,” *Sound and vibration*, pp. 54–59, 2007.
- [8] A. D. Helfrick, *Electrical Spectrum and Network Analyzers: A Practical Approach*. Academic Press, 2012.
- [9] S. F. Adam, “Microwave Instrumentation: An Historical Perspective,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 32, no. 9, pp. 1157–1161, 1984.
- [10] W. A. Rytand and D. R. Gildea, “Network Analysis in the Range 100 kHz to 110 MHz,” *Hewlett-Packard Journal*, 1969.
- [11] L. R. D’Addario, “Computer-Corrected Reflectometer Using the HP-8410 and an Apple 11,” *Nut. Radio Astronomy Observatory Electron. Div. Internal Rep*, no. 228, 1982.
- [12] J. D. Terry and R. R. Kunath, “Using A Modified Hewlett Packard 8410 Network Analyzer as an Automated Farfield Antenna Range Receiver,” in *Antenna Measurement Techniques Association Meeting*, no. E-5922, 1990.
- [13] *HP 8566B Spectrum Analyzer 100 Hz to 22 GHz*, Hewlett-Packard, March 1973, [User Manual](#).
- [14] *User’s Guide: Agilent Technologies 8560 E-Series and EC-Series Spectrum Analyzers*, Agilent Technologies, November 2000, [User Manual](#).

- [15] T. Hoppin, “Strategies for Mitigating Risk Related to the Obsolescence of HP 8566/68 Spectrum Analyzers in ATE Systems,” in *Proceedings, IEEE AUTOTESTCON*. IEEE, 2002, pp. 608–620.
- [16] J. Wolle, “Replacing Obsolete Instrumentation—Lessons Learned,” in *2010 IEEE AUTOTESTCON*. IEEE, 2010, pp. 1–5.
- [17] V. Iglesias, J. Grajal, M. A. Sánchez, and M. López-Vallejo, “Implementation of a Real-Time Spectrum Analyzer on FPGA Platforms,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 338–355, 2014.
- [18] *Spectrum Analyzer RF Section*, Hewlett-Packard Company, August 1974, [Operating and Service Manual](#).
- [19] A. I. Harris, “Heterodyne Spectrometers with Very Wide Bandwidths,” in *Millimeter and Submillimeter Detectors for Astronomy*, vol. 4855. SPIE, 2003, pp. 279–289.
- [20] C. Rauscher, V. Janssen, and R. Minihold, *Fundamentals of Spectrum Analysis*. Rohde & Schwarz, 2007, vol. 25.
- [21] K. Rovers, “Front-End Research for a Low-Cost Spectrum Analyser,” Ph.D. dissertation, University of Twente, 2006.
- [22] *RSA2203A & RSA2208A 3 GHz & 8 GHz Real-Time Spectrum Analyzers*, Tektronix, March 2006, [User Manual](#).
- [23] A. Al Adnani, J. Duplacy, and L. Philips, “Spectrum Analyzers Today and Tomorrow: Part 1 Towards Filterbanks-Enabled Real-Time Spectrum Analysis,” *IEEE Instrumentation & Measurement Magazine*, vol. 16, no. 5, pp. 6–11, 2013.
- [24] *Technical Manual: Plug-in Unit, Electronic Test Equipment PL1388/U (Hewlett-Packard Model 8552B)*, Hewlett-Packard Company, September 1978, [Technical Manual](#).
- [25] *XR-2206 Monolithic Function Generator Rev 1.04*, Exar, February 2008, [Technical Data](#).
- [26] *STM32’s ADC Modes and Their Applications*, ST Electronics, March 2010, [Application Note](#).
- [27] H. Sebastian, “HP141T LCD Upgrade,” [GitHub Repo](#). Last Accessed 21 May 2025., 2024.

Appendix A

Appendix

A.1 Data Acquisition Subsystem Code

A.1.1 DAS Macros

The [Data Acquisition Subsystem \(DAS\)](#) is implemented on the STM32H723ZG microcontroller using embedded C and the STM32 HAL library to sample the horizontal, vertical, and pen-lift signals via ADC1 pins PF10, PF9, and PF8, respectively. A 3-channel [DMA](#) stream is used to continuously transfer 16-bit ADC samples into a 402-element buffer, with the [DMA2_Stream0_IRQHandler](#) managing buffer events via HAL [DMA](#) interrupt handling.

```
1  /* ADC Channels */
2  #define ADCx_CHANNEL          ADC_CHANNEL_8 // V_x
3  #define ADCx_CHANNEL2         ADC_CHANNEL_7 // V_y
4  #define ADCx_CHANNEL3         ADC_CHANNEL_6 // V_z
5
6  /* DMA Configuration */
7  #define ADCx_DMA_CHANNEL      DMA_REQUEST_ADC1 // Master ADC1
8  #define ADCx_DMA_STREAM        DMA2_Stream0
9  #define ADCx_DMA IRQn          DMA2_Stream0_IRQHandler
10 #define ADCx_DMA_IRQHandler    DMA2_Stream0_IRQHandler
11
12 ADC_HandleTypeDef AdcHandleMaster;
13 ADC_HandleTypeDef AdcHandleSlave;
14
15 #define ADC_BUF_SIZE 802 // 802 samples (401 samples/channel)
16 volatile uint16_t adc_buffer[ADC_BUF_SIZE] = {0}; // 16-bit for interleaved data
17
18 volatile int adc_error_flag = 0;
19
```

Listing A.1: Showing macros for the [ADC](#) channels, [DMA](#) and the associated interrupts for handling streaming.

A.1.2 Implementing Direct Memory Access for DAS

The `adc_init` function configures ADC1 and ADC2 in double-interleaved mode for continuous 16-bit sampling across three channels, with DMA handling enabled for real-time data acquisition. Although the system is configured to output 32-bit words, the use of a `uint16_t` buffer relies on the HAL to extract the lower 16 bits of each sample.

```

1 int adc_init(void) {
2     ADC_ChannelConfTypeDef sConfig;
3     ADC_MultiModeTypeDef multiMode;
4     //...
5     /* Configure ADC2 (Slave) */
6     AdcHandleSlave.Instance = ADCx_SLAVE;
7     AdcHandleSlave.Init = AdcHandleMaster.Init; // Same config as master
8     if (HAL_ADC_Init(&AdcHandleSlave) != HAL_OK) {
9         adc_error_handler();
10    return -1;
11 }
12
13     /* Configure double-interleaved mode */
14     multiMode.Mode = ADC_DUALMODE_INTERL_FAST; // Fast interleaved mode
15     multiMode.DualModeData = ADC_DUALMODEDATAFORMAT_32_10_BITS;
16     multiMode.TwoSamplingDelay = ADC_TWOSAMPLINGDELAY_1CYCLE;
17     if (HAL_ADCEx_MultiModeConfigChannel(&AdcHandleMaster, &multiMode) != HAL_OK) {
18         adc_error_handler();
19         return -1;
20     }
21     //...
22 }
23

```

Listing A.2: Code snippet of the `adc_init` function which configures the double-interleaved operation of the ADCs.

A.2 Digital Processing Subsystem Code

A.2.1 Using Piecewise Cubic Polynomial Interpolation to Smooth Display

The proposed design implements the Cubic spline interpolation algorithm which uses piecewise cubic polynomials to define a curve within different segments of the data. The code snippet in listing A.3 illustrates how the frequency domain plot of the signal is smoothed to reduce sharp edges that can occur when using linear interpolation. The program plots the CRT-like background grid on the touchscreen, allowing users to analyse the input signal in the tuning section.

```

1  from scipy.interpolate import CubicSpline
2  #...
3  # Perform cubic spline interpolation
4  def interpolate_data(time_ms, power_db):
5      cs = CubicSpline(time_ms, power_db)
6      time_smooth = np.linspace(time_ms[0], time_ms[-1], 1000) # Increase resolution
7      power_smooth = cs(time_smooth)
8      plt.clf()
9      ax = plt.gca()

10
11     # Plot with CRT styling using spline-interpolated data
12     ax.plot(time_smooth, power_smooth, color='limegreen', linewidth=2)

13
14     # CRT-like background and grid
15     ax.set_facecolor('black')
16     fig.patch.set_facecolor('black')
17     ax.grid(True, color='gray', linestyle='--', linewidth=0.5)

18
19     # Grid divisions: 10 (time), 8 (amplitude)
20     time_max = time_ms[-1]
21     ax.set_xticks(np.linspace(0, time_max, 11))
22     ax.set_yticks(np.linspace(0, 80, 9))

23
24     # Labels and title
25     ax.set_xlabel('Frequency (GHz)', color='white', fontsize=12)
26     ax.set_ylabel('Amplitude (dB)', color='white', fontsize=12)
27     ax.set_title('HP141T Spectrogram', color='white', fontsize=14)
28     ax.tick_params(axis='both', colors='white')

29
30     # Set axis limits
31     ax.set_xlim(0, time_max)
32     ax.set_ylim(0, 80)

33
34     canvas.draw()
35     logging.info("Frequency domain plot updated successfully with spline interpolation")
36

```

Listing A.3: Code snippet for performing polynomial interpolation in the DPS to the spectrogram a smoother appearance.

A.3 Graphical User Interface Code

A.3.1 Pause Functionality Implemented on the Single-Board Computer

The code snippet below is used to flag the STM32H723ZG connected to the single-board RPI 4B to pause data sampling. The code also illustrates enabled logging for debugging and getting feedback on the user's actions.

```

1 # Global pause flag
2 pause_flag = False
3
4 def pause_recording():
5     global pause_flag
6     pause_flag = True
7     status_label.config(text="Recording Paused")
8     logging.info("Recording paused by user")
9

```

Listing A.4: Code snippet for function associated with the pause function in latex.

A.3.2 Enabling the Peak Hold Mode (PHM), Average Mode (AvM) and Raw Mode (RwM) Display Modes

The PHM tracks and displays the maximum value seen for each data point and is updated per scan. The `hp141t` Python module includes a `SpectrumProcessor Class` for updating the `time_data` with the maximum value at each index.

```

1 # Update mode-specific data
2 if self.mode == 'PHM':
3     np.maximum(self.peak_data, power_db, out=self.peak_data)
4     return self.peak_data
5 elif self.mode == 'AvM':
6     if self.scan_count == 1:
7         self.avg_data = power_db
8     else:
9         self.avg_data = (self.avg_data * (self.scan_count - 1) + power_db) / self.scan_count
10    return self.avg_data
11 else: # RwM
12     self.raw_data = power_db
13     return self.raw_data
14

```

Listing A.5: Code snippet for enabling the PHM when the user presses the PHM button on the screen.

A.3.3 System Documentation and Tutorial for Users

A `HELP` button is included to enhance the usability of the system, in line with SRO6. The tutorial is displayed in a read-only `Text` widget and explains the system to users.

```

1 def show_help():
2     """Display a tutorial window"""
3     help_window = tk.Toplevel(root)
4     help_window.title("Help - HP141T Spectrogram Tutorial")
5     help_window.geometry("400x300")
6     help_window.configure(bg='black')
7
8     # Tutorial text
9     tutorial_text = """
10    === HP141T Spectrogram Tutorial ===
11
12    Welcome to the HP141T Spectrogram Display!
13    This system visualizes data from the HP141T spectrum analyzer on a touchscreen LCD.
14
15    **Purpose:**
16    - Displays time-domain spectrogram data (0 to 80 dB amplitude, time sweep).
17    - Replicates the CRT display with a green trace and grid.
18
19    **Modes:**
20    - **Peak Hold Mode (PHM):** Shows the largest value seen at each time point, updated
21      ↪ per scan.
22    - **Average Mode (AvM):** Displays the running average of values at each time point,
23      ↪ updated per scan.
24    - **Raw Mode (RwM):** Shows the latest value, overwriting previous data during each
25      ↪ scan.
26
27    **Usage:**
28    - **Mode Selection:** Tap 'Peak Hold Mode', 'Average Mode', or 'Raw Mode' to switch
29      ↪ display modes.
30    - **Pause:** Tap 'Pause Recording' to stop updates.
31    - **Help:** Tap 'Help' to view this tutorial.
32    - Navigate using touchscreen gestures if supported by future updates.
33
34    **Notes:**
35    - Data is loaded from 'spectrogram_data_X.csv' files in /home/pi.
      - Contact support for issues (log at /home/pi/spectrogram_display.log).
36
37    Close this window to return to the display.
38    """

```

```
36  
37     # Create and configure text widget  
38     text_widget = tk.Text(help_window, bg='black', fg='white', font=('Courier', 10), wrap=  
39     ↪ 'word')  
40     text_widget.insert(tk.END, tutorial_text)  
41     text_widget.config(state='disabled') # Make read-only  
42     text_widget.pack(expand=True, fill='both', padx=10, pady=10)
```

Listing A.6: Code snippet for creating a pop-up window with a tutorial when the user interacts with the `HELP` button.

Appendix B

Appendix

B.1 Project Management Documentation

B.1.1 Gantt Chart

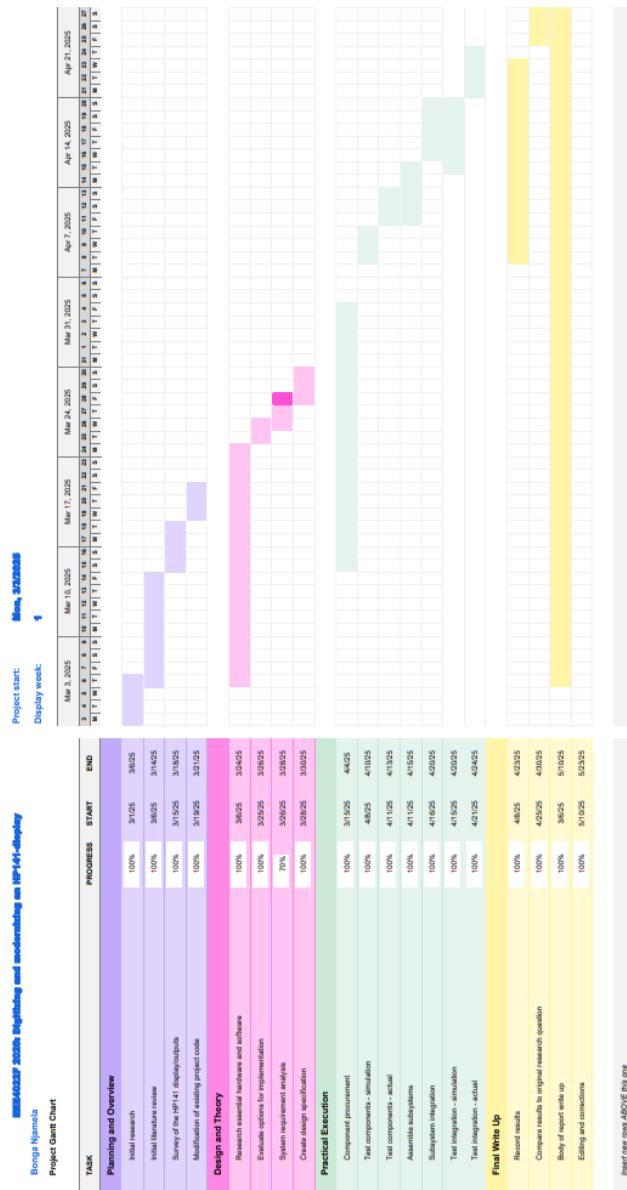


Figure B.1: Gannt Chart employed as a project management tool.

B.1.2 Bill of Materials

EEE4022F
Research
Project
Bongo Njamela,
NJMLUN002

Bill of Materials

*DO NOT ORDER IN RED							
ITEM #	QTY	MANUFACTURER	PRODUCT NUMBER	DESCRIPTION	VENDOR	PRICE PER UNIT	Designation
1	1	Raspberry Pi Ltd	RASPBERRY-PI-4-B-3G	Model 4B Single-board Computer	RS Components	1,860.07	RPI
2	2	Vero Technologies	03-0022	Single-Sided Stripboard 121.92 x 101.6mm	RS Components	27.05	----
3	1	STMicroelectronics	NUCLEO-H723ZG	STM32 Nucleo-144 development board	RS Components	673.77	MCU
4	1	DFRobot	DFR0670	Raspberry Pi LCD Touchscreen	RS Components	1,316.16	LCD
5	1	Texas Instruments	NE555P	555 Timer	RS Components	3.79	U1
6	1	Onsemi	2N2222A8U	PNP BJT Transistor	RS Components	0.93	U2
7	3	Bourns	3362P-5-501LF	Trimmer 5k Potentiometer	RS Components	21.87	R1, R2, R3
8	1	Kemet	C322C204K5R5TA	100 nF Ceramic Capacitor	RS Components	5.37	C1
9	1	Kemet	C315C209B5G5TA	1 pF Ceramic Capacitor	RS Components	0.26	C2
10	1	TDK	FG80CeG3A220INT0	22 nF Ceramic Capacitor	RS Components	15.26	C3
11	1	TE Connectivity	CRF851470R	470R Carbon Film Resistor	RS Components	1.09	R4
12	1	RS Pro	797-7704	2.7k Carbon Film Resistor	RS Components	2.59	R5
13	4	RS Pro	797-8300	10k Carbon Film Resistor	RS Components	3.39	R6, R7, R8, R9
14	3	RS Pro	799-7959	20k Carbon Film Resistor	RS Components	0.23	R10, R11, R12
15	1	RS Pro	797-8943	68k Carbon Film Resistor	RS Components	4.43	R13
16	4	RS Pro	797-8940P	100k Carbon Film Resistor	RS Components	26.23	R14, R15, R16, R17
17	1	Yageo	HHV1W5J-52-200K	200k Metal Film Resistor	RS Components	2.23	R18
18	2	Texas Instruments	LM358AP	Op Amp, 100kHz	RS Components	7.03	U3, U4
19	3	Amphenol	BD252H5-NPPjG-50	Straight Coaxial BNC Adapter	RS Components	80.89	J1, J2, J3
20	1	Raspberry Pi Ltd	TE716DV-R5	Raspberry Pi Power Supply 1.5 m EU Plug	RS Components	199.74	----
21	1	RS Pro	254-3994	100MHz Handheld Oscilloscope	RS Components	16,162.79	----
22	3	Fairchild Semiconductor	2N4148	Diode	RS Components	0.26	D2, D3, D4
23	1	Kingbright	L-34GBC-D	Blue LED	RS Components	5.77	D5
24	1	Energizer	7538900410297	9V Battery Cell	RS Components	226.40	BT3
25	1	RS Pro	285-4728	9V Battery Holder	RS Components	28.43	BT4u
26	1	Chemi-Con	KTD500B47Mg0AuBox	47 uF Multilayer Ceramic Capacitor	RS Components	442.06	C1
27	4	Nichicon	USR1C2010MOD	10 uF Polar Capacitor	RS Components	3.34	C4, C8
28	2	Nichicon	ULD2G010MED	1 uF Polar Capacitor	RS Components	8.81	C2, C6
29	2	Kemet	C340C204K5R5TA	1 uF Capacitor (Non-polarized)	Digikey SA	14.88	C3, C7
30	2	RS Pro	741-0977	1000uF Electrolytic Capacitor	RS Components	5.00	C9, C10
31	1	Kemet	C320C223K5R5TA	22 nF Ceramic Capacitor	RS Components	6.75	C5
32	1	Kemet	C322C204K5R5TA	0.1 uF Ceramic Capacitor	RS Components	4.34	C9, C10
33	1	Myrra	44202	230V AC->23V AC Transformer	RS Components	377.54	T1
34	4	DiodesZetex	2N4007-T	Rectifier Diode 2-Pin DO-41 1000V	RS Components	0.89	D1, D2, D3, D4
35	1	Kingbright	L-536D	Green LED	RS Components	0.79	D5
36	3	Amphenol	BD252H5-NPPjG-50	Straight Coaxial BNC Adapter	RS Components	80.89	J1, J2, J3
37	1	Wurth Elektronik	691211030003	Terminal Block Wall Plug	RS Components	32.32	P2
38	1	Onsemi	PN2222ATA	PNP BJT Transistor	RS Components	5.02	D1
39	3	RS Pro	797-6666	1k Resistor	RS Components	2.53	R1, R2, R25
40	1	TE Connectivity	LR2F1K1	1.2k Resistor	RS Components	1.47	R4
41	1	RS Pro	797-6708	1.2k Resistor	RS Components	3.81	R3
42	5	RS Pro	799-8806	5.1k Resistor	RS Components	0.25	R6, R8, R20, R31, R32
43	10	RS Pro	797-8300	10k Resistor	RS Components	0.33	R9, R11, R21, R22
44	1	Vishay	PT020000201001A100	10k Resistor	RS Components	0.34	R5
45	1	RS Pro	799-8809	4.7k Resistor	RS Components	0.22	R15
46	2	TE Connectivity	CRF851470R	220R Resistor	RS Components	0.78	R16

Figure B.2: Part of the Bill of Materials for tracking component purchases.

B.1.3 GitHub Log for Version Control

```

Completed Signal Conditioning PCB design.

commit af3bef4cf01732378be2c22aa9a3e62e8f787a23
Author: Bonga Njamela <njmlun002@myuct.ac.za>
Date:   Tue Apr 1 20:37:09 2025 +0200

Completed history section in literature review.

commit dde462536b775a0416495317a7473cdf2eae91db
Author: Bonga Njamela <njmlun002@myuct.ac.za>
Date:   Mon Mar 17 03:32:21 2025 +0200

Completed requirements review and system block diagram.

commit 6febbc1e15ad35aea2193d47f16e781539103382
Author: Bonga Njamela <njmlun002@myuct.ac.za>
Date:   Sat Mar 15 02:50:01 2025 +0200

Started comparison of 141T to modern SAs in requirements review.

commit 56914221f7bfbcfa66e8eb80226875326737ae52
Author: Bonga Njamela <njmlun002@myuct.ac.za>
Date:   Thu Mar 13 03:24:01 2025 +0200

Complete methodology overview and start of requirements review.

commit cb57f9ea4dafbfa5011df5119d0a9470c9874c97
Author: Bonga Njamela <njmlun002@myuct.ac.za>
Date:   Wed Mar 12 15:40:42 2025 +0200

Added report as backup and for version control.

commit c30c70353bfa6912adf86b5be99643dde38fc407
Author: Bonga Njamela <75367761+BongaNjamela001@users.noreply.github.com>
Date:   Sat Mar 8 02:23:26 2025 +0200

Initial commit

```

Figure B.3: Snippet of git log since the first commit.

Appendix C

Appendix

C.1 AI Usage History

C.1.1 Prompts and Results for Clarification

What is meant by the scan time base and how is it related to the horizontal output of the IF section of the HP141T

Great question! Let's unpack that clearly.

★ What is the Scan Time Base?

In a spectrum analyzer, the **scan time base** refers to the **timing signal that controls the horizontal sweep** of the display—essentially telling the display how fast to move from the left side of the screen to the right as frequencies are swept. It's the **time-axis driver** of the frequency sweep.

In analog spectrum analyzers like the **HP141T**, this is **not just a display control**—it's tightly integrated into the actual signal measurement process.

⚙ Relation to the Horizontal Output of the IF Section in the HP141T

In the HP141T system:

- The **IF section** (such as the 8553B RF section and 8552B IF section) processes the incoming signal and tunes through a range of frequencies.
- As it tunes (sweeps), the **Local Oscillator (LO)** in the RF section sweeps across a frequency range.
- Simultaneously, a **ramp voltage** is generated, which corresponds to that sweep.

Figure C.1: AI was used for clarifying terms regarding the HP141T system before research was performed.

Can you explain how the -5V to +5V triangular how the output is related to the scan time base between 0.1 ms to 10 s? Does the duty cycle affect the display?

Absolutely! Let's break this down in the context of the HP141T and traditional analog spectrum analyzers.

▲ Triangular Scan Voltage and Scan Time Base

In spectrum analyzers like the HP141T, the **scan time base** is generated using a **ramp** or **triangular waveform**, typically ranging from **-5 V to +5 V**. This waveform is critical for two main things:

1. Controlling the Frequency Sweep of the LO (Local Oscillator)
2. Controlling the Horizontal Sweep on the CRT Display

In a **triangular wave**:

- The **rising edge** controls the **forward sweep** (e.g., from lower frequency to higher frequency).
- The **falling edge** often corresponds to a **retrace period**, where either:
 - Nothing is displayed (i.e., beam is blanked), or
 - A reverse sweep is shown (some analyzers can do this for bidirectional sweeps).

Figure C.2: AI was used for clarifying terms regarding the HP141T system before research was performed.

C.1.2 Prompts for Editing

How would you rephrase the following in the introduction to the experimental test procedures section "the following section explores the setup employed"

Here are a few rephrased alternatives for "the following section explores the setup employed," suitable for the introduction to the experimental test procedures section:

1. "This section outlines the experimental setup used for testing."
2. "The subsequent section describes the configuration implemented during the experiment."
3. "This section details the setup and procedures used in the experimental testing."
4. "An overview of the test setup and methodology is provided in this section."
5. "This section presents the setup employed in conducting the experimental procedures."

Would you like the tone to be more formal, more concise, or more explanatory depending on the rest of your writing style?

Figure C.3: AI was used to restructure phrases.

C.1.3 Prompts for Debugging

Appendix D

Appendix

D.1 Graduate Attributes

D.1.1 GA Table