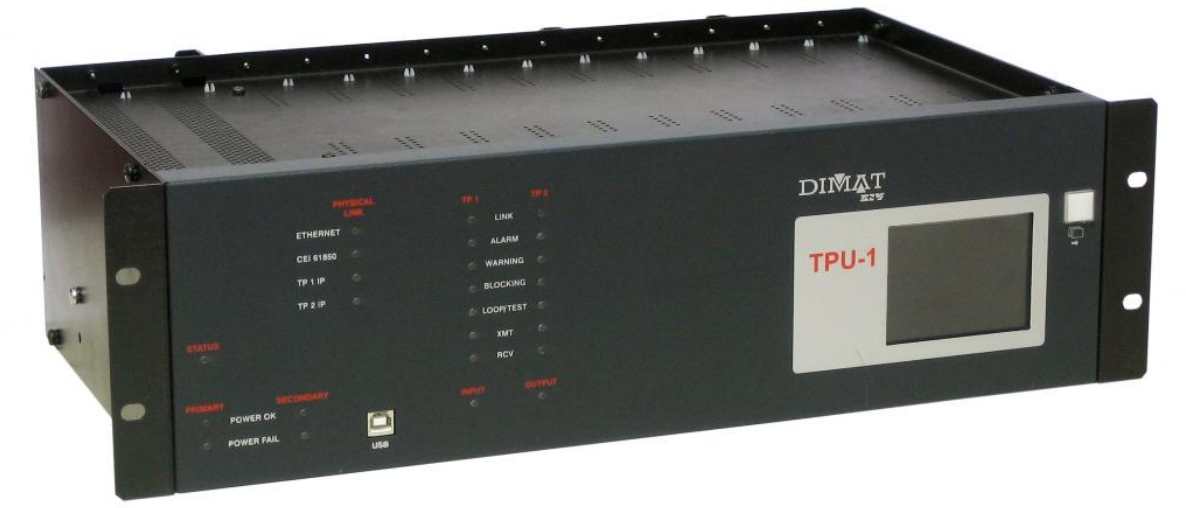

Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment in Electrical Substations

Sergio Bret Villalta

Project Proposal and Work Plan

REVISION HISTORY AND APPROVAL RECORD

Revision	**Date**	**Purpose**
0	03/10/2025	Document creation
0	06/10/2025	Document revision - Entity tutor
0	06/10/2025	Document revision - UPC tutor

DOCUMENT DISTRIBUTION LIST

Name	E-mail
[Sergio Bret Villalta]	sergio.bret@estudiantat.upc.edu
[Viente Jimenez Serres] - UPC tutor	vicente.jimenez@upc.edu
[David Gil Donate] – Entity tutor	DAVID.GIL@ZIVAUTOMATION.COM

WRITTEN BY: Sergio Bret Villalta Date		REVIEWED AND APPROVED BY: Date	
Name	Sergio Bret Villalta	Name	
Position	Project author	Position	Project Supervisor

0. CONTENTS

1. PROJECT OVERVIEW

This project stems from the previous implementation of a series of automated tests for a ZIV TPU-1 Teleprotection Unit (TPU) used in electrical substations. In an initial phase, a set of Python scripts for functional testing, using the Robot Framework and Selenium library, and a Graphical User Interface (GUI) for control is established to automate the configuration of the equipment's web interface.

The main objective of this Final Degree Project is to expand this previous framework, going beyond the simple functional verification of the equipment's web configuration by validating its robustness under more realistic operating conditions. This will be achieved by incorporating Hardware-in-the-Loop (HIL) tests to validate its physical modules, such as the IPTU contact interface. The project will attempt to simulate in a laboratory setting some of the noise conditions that occur in a hostile electromagnetic environment, such as that of electrical substations.

The final deliverable will be a fully integrated system, combining the Python/Robot Framework software with a Raspberry Pi-based controller, that allows for the repeatable and reliable evaluation of the TPU's performance limits as defined by standards like IEC 60834-1. This will not only identify potential design weaknesses but also establish a robust validation protocol, ultimately contributing to the enhanced safety and reliability of critical electrical infrastructure.

2. INTRODUCTION AND MOTIVATION

The security and stability of electrical grid infrastructures largely depend on the proper functioning of the protection, monitoring, and control equipment. Among these, we find the ZIV TPU-1 Teleprotection Units (TPUs), which play a very important role, as they must be able to act quickly and reliably in the face of any anomaly or problem on the grid. A critical factor in the design of these devices is their robustness against environments that we can consider more electromagnetically hostile. The scope of work for this project will be that of electrical substations, which, due to the high levels of noise and disturbances usually present, can compromise the proper functioning and performance of the TPUs.

For this project, we will start from a foundation that I developed during the curricular internship I carried out last semester. It consists of a base of automated functional tests, implemented as a set of Python scripts using the Robot Framework and Selenium library, and a first version of a GUI environment where some of these can be executed in an easy and accessible manner. However, the fact that they are “functional” means that they are limited to verifying the configuration via the equipment's integrated web server and operation of the equipment under ideal laboratory conditions. It doesn’t evaluate the device at a hardware level, for example, the physical response of its IPTU modules, under the stress that would exist in a real scenario. Therefore, it does not validate its robustness.

This final degree project aims to overcome this limitation. The proposal consists of designing and developing a complete automated Hardware-in-the-Loop (HIL) testbench that allows for the simulation of the adverse conditions typical of an electrical substation, in a controlled manner and with the possibility of performing multiple repetitions. This implies that we will have to go through two phases. On one hand, the evolution of the current control and test execution software to one that features a more professional and advanced environment, with the capability for automatic report generation. On the other hand, the design and construction of the necessary hardware to generate and inject noise into the equipment under test, as well as to physically trigger its command inputs.

The main motivation for this project is, therefore, twofold. Firstly, it seeks to increase the quality and reliability of the final product by identifying possible bugs or failures before it reaches the market, allowing for the evaluation of the TPU's performance limits. Secondly, it will aim to provide the company, specifically the R&D and application department, with a strategic tool with which time and costs can be saved in the testing of future equipment and modules. In conclusion, we will be able to provide lasting value, ensuring that the teleprotection equipment complies with the reliability standards, such as IEC 60834-1 and ANSI C37.90.1, that this type of infrastructure demands. (1)

3. OBJECTIVES

The main objective of this Final Degree Project is the design and development of a fully automated test bench for the validation of the robustness of ZIV's TPU-1 teleprotection equipment, through the simulation and control of the adverse electromagnetic conditions typical of electrical substations.
Specifically, to achieve this main objective, we will need to break it down into the following specific objectives, which can be grouped into the different development phases of the project:

3.1. *Phase 1: Development and Improvement of the Control and Testing Software.*

- To evolve and professionalize the current graphical user interface (GUI) and its underlying Python-based test environment, which uses Robot Framework and Selenium, turning it into a complete control panel for the management and execution of both the already implemented web-based tests and the future.

- To implement tools for real-time data storage and visualization within the GUI, to allow for the monitoring of key parameters during tests and trials, such as SNMP responses and chronological log events. (8)

- To implement the automatic generation of reports, with a standardized format, of the results obtained from the robustness tests, as already done by Robot Framework for functional tests. This report must be able to detail the test configuration and scenario, the data collected, and the final verdict, PASS/FAIL.

3.2. *Phase 2: Design and Construction of the Hardware for robustness testing.*

- To design and specify the necessary hardware for the test bench. It must be capable of generating and injecting controlled electrical disturbances into the communication signals of the equipment under test, in line with standards such as IEC 61000-6-5 and ANSI C37.90.1. (1)(4)

- To build a functional prototype of the testbench. Precise synchronization between the control signals and the reception of measurements will be critical to obtain correct results. Signal transmission delays will have to be considered throughout all processing, both in software and hardware.

- To design a device, based on a Raspberry Pi 4 and 8-channel relay modules, for sending hardware input commands to the IPTU protection interface modules. This way, we will replace the current method, which is based on the software simulation available in the TPU's processing module (MWTU). (6) (3)

3.3. *Phase 3: Integration, Validation, and Analysis*

- To integrate the designed hardware system, specifically the Raspberry Pi controller, with the graphical software control interface and the Robot Framework scripts. We'll need to ensure stable and fluid communication between them, for example, via the pyserial library.

- To define and implement a set of relevant HIL robustness tests that simulate real-world scenarios of an electrical substation environment, verifying the complete command chain from a software trigger to a physical input activation.

- To execute the tests and analyse the obtained results. This way, we will be able to identify possible bugs or glitches, some of the equipment's operational limits, especially regarding command transmission times under stress, and most importantly, we will be able to provide recommendations for improvement.

3.4. *Optional / Desirable Objective*

- To migrate the entire test environment, including the GUI, Robot Framework engine, and all Python scripts, to the Raspberry Pi. The goal is to transform the test bench into a fully standalone and portable test station that does not depend on an external PC for its operation.

4. STATE OF THE ART

4.1. *Electromagnetic Compatibility (EMC) and Standards*

Electrical substations are environments with high electromagnetic interference (EMI). This means that the installed equipment must comply with strict electromagnetic compatibility (EMC) standards. The TPU-1 manual specifies compliance with these standards, such as:

- **IEC 61000-6-5** (1): Generic immunity standard for power plant and substation environments. Some descriptions of these electromagnetic phenomena that may be useful in our project are:

  - Electrical Fast Transients/Bursts (EFT/B)
  - Surges
  - Radiated, radiofrequency, electromagnetic fields (RF)
  - Conducted disturbances, induced by radio-frequency fields.

  This standard also specifies the test and severity levels, such as the voltage, current, or field strength levels for each disturbance.
  We can also find the performance acceptance criteria. All of this will be of relevant importance to our project, as our test bench will have to try to recreate a simplified version of these phenomena. For example, the noise generation hardware could be designed to generate an RF signal on a communication cable to see how it affects the TPU-1.

- **IEC 60834-1** (1) (2): Performance and testing standard for teleprotection systems. Unlike the previous standard (which focused more on the "causes"), this one speaks of the "consequences." That is, how the equipment must function despite the presence of noise. Among some of the conclusions we have reached, we find the following:

  - The equipment must operate correctly when there is a real command. It is vital that the noise we inject does not cause a false activation of an output.
  - We must verify that, even with noise, if an input is activated, the command is sent and received correctly.
  - The standard also describes the maximum times for command transmission. One of the objectives will be to measure these times and see if they are affected by the disturbances. (7)
  - It also specifies different performance classes (P1, P2, etc.) that define the probability of admitted errors.

  As we have seen, this standard gives us the key performance indicators (KPIs) that we must measure. This means that our system will not only have to inject noise, but it will also have to verify if, during the test, the TPU-1 continues to comply with the security, reliability, and time criteria that this standard demands.

- **ANSI C37.90.1/C37.90.2** (1): US standards for substation equipment and for relays. They are related to the ability to withstand surges and radio frequency emissions. Some of the interesting tests described in this standard are:

  For the case of **ANSI C37.90.1** (Surge Withstand Capability):
  - **Damped oscillatory wave:** A radio frequency wave that gradually attenuates is injected. It simulates the oscillations that occur when opening or closing disconnectors.

    Fast transient (EFT): Very similar to the EFT/B test of IEC 61000, but with its own specifications.

  For the case of ANSI C37.90.2 (Radiated Radio Frequency Interference):
  - It defines a robustness test against radiated electromagnetic fields, specifying a severity level. It is especially relevant for simulating the effect of "walkie-talkies" or other radio emitters near the equipment.

  These standards give us examples of disturbance signals that we could try to generate with our hardware.

  (1)(4)(9)

Our project's test bench will have to establish a space to verify the robustness of the TPU-1 against phenomena like those described in these standards.

4.2. *Equipment Under Test: Universal Teleprotection TPU-1*

ZIV's TPU-1 is a modular, flexible, and universal teleprotection equipment. Its architecture is based on a 19" chassis, with 13 slots where different modules are connected: (1)(3)(4)

- **Base modules:** Such as the power supply (ATPU) and the main processing module (MWTU), which includes a web server for the management and monitoring of the equipment. (1) (3)

- **Protection interfaces:** They allow communication with the relays. They can be analog by contacts (IPTU module) or digital, compatible with the IEC 61850 standard through GOOSE messages (IEPT or KWTU modules). (1) (5)

- **Line interfaces:** They offer a wide variety of communication channels, such as digital electric (IETU, IDTU), optical (IOTU, IOCT, IOCS), for packet networks (IPIT), or analog by tones (IBTU). (1) (3)

This project focuses on physically validating the analog interface modules by contacts (IPTU), which are currently only tested at the software configuration level, through the TPU's configuration website.

4.3. *Test Automation Methodologies*

The current state of the project uses a modern approach for software test automation, making use of the following Python libraries:

- **Robot Framework:** It is an open-source test automation framework for acceptance testing and acceptance test-driven development (ATDD). Its structure is keyword-oriented (keywords), which facilitates the creation of high-level functions and tests.

- **Selenium Library:** It is one of the most popular libraries for Robot Framework. It allows controlling a web browser for the automation of web interface tests.

This combination is ideal for testing the configuration of the TPU-1 through its integrated web server. However, there is an obvious problem: this methodology cannot interact directly with the physical world. It cannot close a contact or activate an opto-isolated input of an IPTU module, nor measure the system's response. (6)

4.4. *Conclusions and Project Justification*

The analysis we have carried out makes a clear gap between the capabilities of the current testing system (focused on software and the web UI) and the need for a complete validation of the TPU-1's hardware. While the use of Robot Framework and Selenium is an excellent practice for configuration, it doesn't guarantee that the hardware will respond correctly in a real scenario.

This project, therefore, justifies the need to build a bridge between the world of software and the world of hardware. The development of an HIL system, controlled by the same Python and Robot

Framework platform, will create an integral, flexible, and low-cost testing solution, perfectly adapted to the company's needs, that will be able to guarantee the reliability of its equipment.

5. METHODOLOGIES

As we have mentioned previously, this project addresses different foundations, combining software with hardware design and systems integration. For this reason, we will employ different methodologies in each phase:

- **For Software Development (Phase 1):** We will use an iterative approach, based on incremental improvement cycles. This method will allow us to evolve the software program that I already developed in a controlled manner, adding new functionalities (such as external hardware control), testing them in isolation, and progressively integrating them into the graphical user interface (GUI).

- **For Hardware Development (Phase 2):** We will follow a more classic model called "waterfall". It is ideal for physical design and consists of well-defined sequential phases: requirements specification, circuit design, component acquisition, prototype assembly, and hardware unit tests.

- **For Integration and Validation (Phase 3):** We will apply a systems integration methodology, where the software and hardware components, which we will develop and test separately, will be joined to form the final test bench. With the latter, we will validate the correct communication and synchronization between all parts of the system.

5.1. *Tools and Technologies*

To choose the tools we will work with, we have tried to find a continuation with those we already used in the foundations of this project, when we developed the first version of the testing software. In addition, we have sought a balance between power and flexibility.

- **Software Environment:**
    - **Programming Language:** Following the test base that we already implemented in the previous internship, Python will be the main language.
    - **Testing Framework:** We will continue to use the Robot Framework library as the main orchestrator for the tests, taking advantage of its keyword-based structure to define high-level test cases.
    - **Web Automation:** The Selenium Library will continue to be used for interaction with the TPU-1's web interface, mainly for initial configuration and results verification.
    - **Hardware Communication:** The PySerial library will be used to establish communication between the control computer and the test bench's control unit (Raspberry Pi).
    - **Graphical User Interface (GUI):** We will continue to use the Tkinter library.

- **Hardware Environment:**
    - **Control Unit:** The core of the test bench will be a Raspberry Pi 4 (8GB). Initially, it will function as a dedicated HIL controller, receiving commands from the main PC to activate physical components.
      As a desirable final step (optional objective), it is planned to host the entire software environment, turning it into a standalone test station.

    - Despite the great potential of the Raspberry Pi, we will also make use of an Arduino UNO for the initial prototype phases.

    - **Actuation Interface:** 8-channel relay modules (preferably solid-state) will be used to act on the physical inputs of the TPU-1. These relays ensure electrical isolation between the control unit and the equipment under test. Logic level converters will be included to adapt the 3.3V of the Raspberry Pi to the 5V required by the relay modules.

    - **Equipment Under Test (DUT):** The equipment to be validated will be a ZIV TPU-1 Universal Teleprotection unit, equipped with the relevant modules, especially protection interface modules by contacts (IPTU).

- **Version Control:** We will start using Git for code management, as we did not use it in the first version of the software. This way, we will ensure traceability and collaborative work.

6. WORK PLAN

6.1. *Work Breakdown Structure*

WP1: Development of the Control and Testing Software
WP2: Design and Construction of the HIL Test Bench
WP3: Final Integration and Validation Cycle
WP4: Final review and report

6.2. *Work Packages, Tasks and Deliverables*

Project: **Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment**	**WP ref: 1**	
Major constituent: **Development of the Control and Testing Software**	Sheet 1 of 2	
Short description: This phase focuses on growing the existing test software. The current Python environment, using Robot Framework and Selenium for web UI testing, will be extended to control the new Hardware-in-the-Loop (HIL) system. This requires adapting the GUI, developing new Python libraries to manage communication with the Raspberry Pi controller, and improving the automated reports.	Planned start date: 06-10-2025 Planned end date: 30-10-2025 **137,75h** Start event: 13-10-2025 End event: 30-10-2025	
Internal task T1: Analyse the existing Robot Framework and Selenium codebase. Design and implement new Python libraries and keywords to manage communication (e g. via PySerial) with the Raspberry Pi controller. **Internal task T2**: Modify the current Tkinter graphical user interface to include new controls (buttons, status indicators) for executing Hardware-in-the-Loop tests and visualizing their results in real-time. **Internal task T3**: Define the logic for the new HIL test cases within Robot Framework, structuring the keywords and sequences that will be used to command the hardware and verify the results. **Internal task T4**: Extend the automatic report generation capabilities of Robot Framework to include specific results from physical tests, such as hardware command response times and the PASS/FAIL status of HIL validations.	Deliverables: T1 T2 T3 T4	Dates: 13-10-2025 21-10-2025 24-10-2025 30-10-2025

Project: **Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment**	**WP ref: 2**	
Major constituent: **Design and Construction of the HIL Test Bench**	Sheet 1 of 2	
Short description: This phase covers the design and construction of the Hardware-in-the-Loop testbench. A Raspberry Pi 4 will work as the control unit, managing 8-channel relay modules via its GPIO pins. This setup is designed to physically activate the inputs of the TPU-1's IPTU module, replacing the current software-based simulation.	Planned start date:31/10/2025 Planned end date: 02/12/2025 166,75h Start event: 31-11-2025 End event: 02-12-2025	
Internal task T1: Create the complete circuit and connection diagrams for the HIL system, detailing the wiring between the Raspberry Pi's GPIO pins, logic level converters, 8-channel relay modules, and the input terminals of the TPU-1's IPTU module. **Internal task T2**: Get all necessary components for the project's shopping list (Raspberry Pi, relays, power supplies, etc.) and physically assemble them into a functional test bench chassis. **Internal task T3**: Develop and deploy the Python scripts on the Raspberry Pi. These scripts will create a listener (e g. a serial or socket server, like what we built for the SNMP) to receive commands from the main control PC. They will also manage the GPIO pin activations to control the relay module. **Internal task T4**: Perform independent tests on the assembled HIL bench. Verify that the Raspberry Pi can reliably and activate each relay on command and debug any issues with the electrical connections or controller scripts.	Deliverables: T1 T2 T3 T4	Dates: 06-11-2025 14-11-2025 24-11-2025 02-12-2025

Project: **Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment**	**WP ref: 3**
Major constituent: **Final Integration and Validation Cycle**	Sheet 2 of 2
Short description: In this final phase, the software control panel is integrated with the HIL hardware. We'll execute the complete end-to-end test cases. Commands sent from the GUI will trigger the Raspberry Pi to activate the TPU-1's inputs. The equipment's response will be verified via SNMP and its chronological log, followed by data analysis and documentation. **Internal task T1**: Establish the communication connection between the main control software and the Raspberry Pi controller. Perform end-to-end connectivity tests to ensure commands from the GUI are correctly executed by the hardware. **Internal task T2**: Implement the test cases defined in Phase 1. Execute an understandable test campaign that simulates real-world operations and command sequences on the TPU-1, such as activating inputs and verifying outputs. **Internal task T3**: Collect the automated reports and logs	Planned start date: 03/12/2025 Planned end date: 09/01/2026 174h Start event: 03/12/2025 End event: 09/01/2026

generated by Robot Framework and TPU-1's chronological log. Analyse the performance data, focusing on command execution times, system reliability, and identifying any bugs or operational limits. **Internal task T4 (Optional)**: If time allows after completing all primary objectives, this task involves migrating the full Python software environment (GUI, Robot Framework, Selenium) to the Raspberry Pi. This includes setting up the desktop environment, all dependencies, and browser drivers to make the test bench fully autonomous and independent from the PC.	

	Deliverables:	Dates:
	T1	15-12-2025
	T2	29-12-2025
	T3	09-01-2026
	T4	09-01-2026

Project: **Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment**	**WP ref: 4**	
Major constituent: **Final review and report**	Sheet 2 of 2	
Short description: The final phase will consist of gathering all the knowledge acquired and deliverables, preparing a report about the completed project and presenting the results obtained. Potential recommendations or improvements for the system will be attached.	Planned start date: 12/01/2026 Planned end date: 02/01/2026 65,25h Start event: 12/01/2026 End event: 02/01/2026	
Internal task T1: Prepare a final project report with documentation o the entire project, including the final design of the testbench, the implemented software, the test results, and the analysis. Write the final TFG memory. Internal task T2: Prepare Project Presentation. Internal task T3: Suggest possible future improvements or developments.	Deliverables: T1 T2 T3	Dates: 21-01-2026 22-01-2026 22-01-2026

6.3. *Time Plan (Gantt diagram)*

Project: Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	Time (hours)	2025-10-06	2025-10-07	2025-10-08	2025-10-09	2025-10-10	2025-10-11	2025-10-12	2025-10-13	2025-10-14	2025-10-15	2025-10-16	2025-10-17	2025-10-18	2025-10-19	2025-10-20	2025-10-21	2025-10-22	2025-10-23	2025-10-24	2025-10-25	2025-10-26	2025-1
WP1: Development of the Control and Testing Software	137,75																						
T1: Framework Analysis and Extension	43,5	7,25	7,25	7,25	7,25	7,25			7,25														
T2: GUI Adaptation for HIL Control.	43,5									7,25	7,25	7,25	7,25			7,25	7,25						
T3: HIL Test Case Logic Development.	21,75																	7,25	7,25	7,25			
T4: Reporting System Enhancement.	29																						7,25
WP2: Design and Construction of the HIL Test Bench	166,75																						
T1: Electrical Design and Schematics.	36,25																						
T2: Component Procurement and Physical Assembly.	43,5																						
T3: Raspberry Pi Controller Scripting.	43,5																						
T4: Hardware Unit Testing and Debugging.	43,5																						
WP3: Development of External Testing Hardware	174																						
T1: Software-Hardware System Integration.	58																						
T2: Integrated Test Case Implementation and Execution.	58																						
T3: Data Collection and Performance Analysis	58																						
WP4: Final review and report	65,25																						
T1: Final report with documentation	54,375																						
T2: Prepare Project Presentation.	7,25																						
T3: Suggest possible future improvements or developments.	3,625																						
TOTAL	**543,75**																						

Project: Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	Time (hours)	2025-10-28	2025-10-29	2025-10-30	2025-10-31	2025-11-01	2025-11-02	2025-11-03	2025-11-04	2025-11-05	2025-11-06	2025-11-07	2025-11-08	2025-11-09	2025-11-10	2025-11-11	2025-11-12	2025-11-13	2025-11-14	2025-11-15	2025-11-16	2025-11-17	2025-1
WP1: Development of the Control and Testing Software	137,75																						
T1: Framework Analysis and Extension	43,5																						
T2: GUI Adaptation for HIL Control.	43,5																						
T3: HIL Test Case Logic Development.	21,75																						
T4: Reporting System Enhancement.	29	7,25	7,25	7,25																			
WP2: Design and Construction of the HIL Test Bench	166,75																						
T1: Electrical Design and Schematics.	36,25				7,25			7,25	7,25	7,25	7,25												
T2: Component Procurement and Physical Assembly.	43,5											7,25			7,25	7,25	7,25	7,25	7,25				
T3: Raspberry Pi Controller Scripting.	43,5																					7,25	7,25
T4: Hardware Unit Testing and Debugging.	43,5																						
WP3: Development of External Testing Hardware	174																						
T1: Software-Hardware System Integration.	58																						
T2: Integrated Test Case Implementation and Execution.	58																						
T3: Data Collection and Performance Analysis	58																						
WP4: Final review and report	65,25																						
T1: Final report with documentation	54,375																						
T2: Prepare Project Presentation.	7,25																						
T3: Suggest possible future improvements or developments.	3,625																						
TOTAL	**543,75**																						

Project: Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	Time (hours)	2025-11-19	2025-11-20	2025-11-21	2025-11-22	2025-11-23	2025-11-24	2025-11-25	2025-11-26	2025-11-27	2025-11-28	2025-11-29	2025-11-30	2025-12-01	2025-12-02	2025-12-03	2025-12-04	2025-12-05	2025-12-06	2025-12-07	2025-12-08	2025-12-09	2025-12-1
WP1: Development of the Control and Testing Software	137,75																						
T1: Framework Analysis and Extension	43,5																						
T2: GUI Adaptation for HIL Control.	43,5																						
T3: HIL Test Case Logic Development.	21,75																						
T4: Reporting System Enhancement.	29																						
WP2: Design and Construction of the HIL Test Bench	166,75																						
T1: Electrical Design and Schematics.	36,25																						
T2: Component Procurement and Physical Assembly.	43,5																						
T3: Raspberry Pi Controller Scripting.	43,5	7,25	7,25	7,25			7,25																
T4: Hardware Unit Testing and Debugging.	43,5							7,25	7,25	7,25	7,25			7,25	7,25								
WP3: Development of External Testing Hardware	174																						
T1: Software-Hardware System Integration.	58															7,25	7,25	7,25				7,25	7,25
T2: Integrated Test Case Implementation and Execution.	58																						
T3: Data Collection and Performance Analysis	58																						
WP4: Final review and report	65,25																						
T1: Final report with documentation	54,375																						
T2: Prepare Project Presentation.	7,25																						
T3: Suggest possible future improvements or developments.	3,625																						
TOTAL	**543,75**																						

Project: Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	Time (hours)	2025-12-11	2025-12-12	2025-12-13	2025-12-14	2025-12-15	2025-12-16	2025-12-17	2025-12-18	2025-12-19	2025-12-20	2025-12-21	2025-12-22	2025-12-23	2025-12-24	2025-12-25	2025-12-26	2025-12-27	2025-12-28	2025-12-29	2025-12-30	2025-12-31	2026-01-01	2026-0
WP1: Development of the Control and Testing Software	137,75																							
T1: Framework Analysis and Extension	43,5																							
T2: GUI Adaptation for HIL Control.	43,5																							
T3: HIL Test Case Logic Development.	21,75																							
T4: Reporting System Enhancement.	29																							
WP2: Design and Construction of the HIL Test Bench	166,75																							
T1: Electrical Design and Schematics.	36,25																							
T2: Component Procurement and Physical Assembly.	43,5																							
T3: Raspberry Pi Controller Scripting.	43,5																							
T4: Hardware Unit Testing and Debugging.	43,5																							
WP3: Development of External Testing Hardware	174																							
T1: Software-Hardware System Integration.	58	7,25	7,25			7,25																		
T2: Integrated Test Case Implementation and Execution.	58						7,25	7,25	7,25	7,25			7,25	7,25	7,25					7,25				
T3: Data Collection and Performance Analysis	58																				7,25	7,25	7,25	7,25
WP4: Final review and report	65,25																							
T1: Final report with documentation	54,375																							
T2: Prepare Project Presentation.	7,25																							
T3: Suggest possible future improvements or developments.	3,625																							
TOTAL	**543,75**																							

Project: Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	Time (hours)	2026-01-03	2026-01-04	2026-01-05	2026-01-06	2026-01-07	2026-01-08	2026-01-09	2026-01-10	2026-01-11	2026-01-12	2026-01-13	2026-01-14	2026-01-15	2026-01-16	2026-01-17	2026-01-18	2026-01-19	2026-01-20	2026-01-21	2026-01-22	2026-01-23	2026-0
WP1: Development of the Control and Testing Software	137,75																						
T1: Framework Analysis and Extension	43,5																						
T2: GUI Adaptation for HIL Control.	43,5																						
T3: HIL Test Case Logic Development.	21,75																						
T4: Reporting System Enhancement.	29																						
WP2: Design and Construction of the HIL Test Bench	166,75																						
T1: Electrical Design and Schematics.	36,25																						
T2: Component Procurement and Physical Assembly.	43,5																						
T3: Raspberry Pi Controller Scripting.	43,5																						
T4: Hardware Unit Testing and Debugging.	43,5																						
WP3: Development of External Testing Hardware	174																						
T1: Software-Hardware System Integration.	58																						
T2: Integrated Test Case Implementation and Execution.	58																						
T3: Data Collection and Performance Analysis	58			7,25		7,25	7,25	7,25															
WP4: Final review and report	65,25																						
T1: Final report with documentation	54,375										7,25	7,25	7,25	7,25	7,25			7,25	7,25	3,625			
T2: Prepare Project Presentation.	7,25																			3,625	3,63		
T3: Suggest possible future improvements or developments.	3,625																				3,63		
TOTAL	543,75																						

7. REQUIRED RESOURCES

7.1. *Project Requirements*

7.1.1. Knowledge Requirements

- Programming: Advanced skills in Python, specifically with test automation libraries such as Robot Framework and Selenium, and GUI development with Tkinter.

- Hardware and Electronics: Basic understanding of microcontrollers (Raspberry Pi and Arduino), GPIO management, and interfacing with external circuits, like relay modules.

- Networking: Knowledge of ethernet protocols and monitoring services, like SNMP.

- Equipment Specifics: Familiarity with the architecture and operation of the ZIV TPU-1 and its main modules, especially the MWTU and IPTU.

7.1.2. Hardware Resources

- Equipment Under Test (DUT): A ZIV TPU-1 Univesal Teleprotection unit, equipped with at least, an ATPU (power supply), the MWTU (processing module), IPTU (protection interface) and a pair of communication lines modules.

- Control Unit: A Raspberry PI 4 (8Gb) with its corresponding power supply and SD card. An Arduino UNO will also be used for prototyping.

- Interfacing Hardware: 8 channels relay modules (solid-state), logic level converters (3.3V to 5V), and a dedicated 24V power supply for the TPU.

- Auxiliary Equipment: A development PC, multimeters, and necessary wiring cables.

7.1.3. Software Resources

- Operating System: A Windows OS on the development PC and Raspberry PI OS on the controller.

- Development environment: A Python 3.x environment with libraries including Robot Framework, Selenium Library and Pyserial, and Tkinter.

- Version Control: Git will be used for source code management.

7.1.4. Organizational Resources

- Resource Access: Access to a pair of individual TPU-1 equipment with its technical documentation.

- Support: Regular project meetings and technical support from the Application ZIV Department.

7.2. *Project Specifications*

7.2.1. Functional Specifications

- The system must provide a Graphical User Interface (GUI) for the user to configure, manage and execute the different tests in an intuitive manner.

- The testbench must be able to **physically** activate the command inputs of the TPU-1's IPTU module via the Raspberry Pi and the relay system. This way we'll be able to perform Harware-In-Loop (HIL) tests.

- The system must verify the TPU-1's response by monitoring its status through the SNMP agent and parsing the chronological register.

- The result must integrate smoothly with the existing test system built with Robot Framework.

7.2.2. Software Specifications

- The control program will be a desktop GUI developed in Pyhton using the Tkinter Library.

- The test logic will extend the existing Robot Framework structure, implemented during my previous internship at ZIV. We'll add new keywords for HIL operations.

- The system shall automatically generate test reports in a standardized format (HTML or XML), indicating the results, parameters and PASS/FAIL verdict for each test.

- Communication with Raspberry PI controllers from the host PC must be reliable, implemented via serial or socket-based protocol.

7.2.3. Hardware Specifications

- The HIL controller unit will be a Raspberry Pi 4.

- The connection between the interface to the DUT (IPTU) must use relays to ensure electrical isolation and prevent the risk of damage to the TPU-1 or the control unit.

- The hardware design must be scalable, allowing for future extensions, like an integration of the noise generation and injection module.

7.2.4. Documentation Specifications

- User Documentation: A brief and intuitive manual must be created explaining how to set up the testbench and operate it through the GUI.

- Technical Documentation: The project's source code must be well-commented, and the final TFG report must contain the design and implementation of both the hardware and software.

8. BUDGET

This section provides an estimated budget for the project hardware, software and human resources. The goal is to present an understandable view of the total investment and value of the work.

8.1. *Hardware Costs*

Component	**Description**	**Quantity**	**Unit Price (€)**	**Total Price (€)**
Control Unit	Raspberry Pi 4 Model B (8GB)	1	80,00 €	80,00 €
Control Unit (Proto)	Arduino UNO Rev3 (Official)	1	29,30 €	29,30 €
Power Supply (Pi)	Mean Well RS-15-5 (5V, 3A)	1	12,00 €	12,00 €
Storage	SanDisk Ultra 32GB microSDHC Card	1	8,00 €	8,00 €
Actuation Interface	8-Channel Solid State Relay Module	1	20,08 €	20,08 €
Interfacing	Bidirectional Logic Level Converter Module	1	5,91 €	5,91 €
Power Supply (TPU)	24V 2A 50W Switched Power Supply	1	12,99 €	12,99 €
Wiring	Dupont Jumper Wires (M-F, 40 pcs)	1	6,00 €	6,00 €
Connectors	Banana Plugs 4mm (Set of 4)	2	10,00 €	20,00 €
-	**Subtotal Hardware Costs**	-	-	**194,28 €**

8.2. *Software Costs*

All software and libraries used for this project is open source. So, there aren't direct licensing costs associated.

8.3. *Human Resources Costs*

This cost would represent the dedication of an engineer intern, like me, for the total duration of the TFG.

Resource	Total Hours	Hourly Rate (€)	Total Cost (€)
Junior Engineer (Student)	540	10.00	5400.00

8.4. *Budget Summary*

Category	**Cost (€)**
Hardware Costs	194,28 €
Software Costs	0.00
Human Resources Costs	5.400,00 €
Grand Total	5.594,28 €

9. RISK AND CONTINGENCY PLAN

9.1. Technical Risk: Hardware Component Delays or Failures

- **Risk Description:** The electronic parts (Raspberry Pi, relays, etc.) are delayed, arrive defective, or break during the project.

- **Analysis:**
  - Probability**:** Medium.
  - Impact**:** Medium, since it could stop the hardware construction phase.

- **Preventive Plan:**
  - Order all components at the beginning of Phase 2.
  - Choose reliable suppliers with fast shipping options.
  - Buy duplicates of critical, low-cost parts to have spares.

- **Reactive Plan:**
  - If a shipment is delayed, reorganize the plan to work on software tasks that can be done in parallel.
  - If a part doesn't work, use the spare and order a new one immediately.

9.2. *Technical Risk: Software-Hardware Integration Problems*

- **Risk Description:** The communication between the computer (with the GUI) and the Raspberry Pi doesn't work well, is slow, or is difficult to stabilize.

- **Analysis:**
  - Probability**:** High.
  - Impact**:** High, since if they don't communicate, the entire system won't work.

- **Preventive Plan:**
  - Design a simple communication method from the start.
  - Perform a basic connection test early to ensure both systems can communicate before programming the full logic.
  - Write the code in separate, organized parts to find faults more easily.

- **Reactive Plan:**
  - Extra time has already been allocated in the schedule for this task. This time will be used for debugging.
  - Add log messages to the code to see where communication fails.
  - If the chosen method doesn't work, we must be prepared to switch to an alternative (e g. from serial port to Ethernet).

9.3. *Technical Risk: Unexpected Behavior of the TPU-1 Interface*

- **Risk Description:** The relay system does not activate the TPU-1's IPTU module inputs correctly, due to an electrical issue (voltage, current consumption, signal bounce).

- **Analysis:**
  - Probability**:** Low-Medium, since the TPU is an industrial device, but the interface is custom-built.
  - Impact**:** High, since it could make the test bench useless or, in the worst case, damage the TPU-1.

- **Preventive Plan:**
  - Review the technical specifications of the IPTU module in the TPU-1 manual carefully.
  - Use relays to ensure electrical isolation as a key safety measure.
  - Perform initial tests with a multimeter to confirm the relay provides the correct voltage before connecting it to the TPU-1.

- **Reactive Plan:**
  - If something fails, we'll use an oscilloscope to analyze the signal from the relay.
  - Consult the ZIV tutor immediately, as their knowledge of the equipment is the most important resource for solving this problem.
  - If necessary, modify the circuit.

10. MEETING AND COMMUNICATION PLAN

10.1. *Communication Plan with the Company Tutor (ZIV)*

10.1.1. Objectives:

- To follow up on the technical progress of the tasks.
- To make sure the results match ZIV expectations.
- To get help with hardware and integration problems.
- To resolve questions about the TPU-1 and its modules.

10.1.2. Frequency:

- A short weekly meeting of 10-20 minutes to discuss any issue and the plan for the week.
- A longer meeting every two weeks of 30 -45 minutes for a more detailed technical review and to show the process.

10.1.3. Format

- After each long meeting, I'll take note of what was discussed and the next steps.

10.2. *Communication Plan with the Academic Tutor*

10.2.1. Objectives:

- To ensure the project has the right scope and difficulty for a Final Degree Project.
- To review the project's methodology and planning.
- To get guidance on the structure and writing of the final report.

10.2.2. Frequency:

- One week before each submission to my final degree project repository.
- When doubts arise in the fulfillment of any of the objectives of this communication plan.

10.2.3. Format

- I will present a summary of the progress.
- I will send drafts of the final report for review.
- The focus will be on the academic quality of the work and the structure of the document.

11. BIBLIOGRAPHY

- *TPU-1 Instruction Manual:*
  (1) PAGE 6 (ALCANCE Y APLICACIÓN DEL EQUIPO)
  (2) PAGE 31 (INTERFAZ DE LÍNEA DIGITAL)
  (3) PAGE 8 (CONSTITUCIÓN DEL EQUIPO)
  (4) PAGE 29 (COMPATIBILIDAD ELECTROMAGNÉTICA)
  (5) PAGE 9 (MWTU)
  (6) PAGE 10 (IEPT)
  (7) PAGE 40 (CARACTERÍSTICAS TÉCNICAS)
  (8) PAGE 7 (SUPERVISIÓN MEDIANTE AGENTE SNMP)

- (9) GOOGLE SCHOLAR