





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

Sergio Bret Villalta

---

Project Critical Review

---

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 2 of 18		

## REVISION HISTORY AND APPROVAL RECORD

Revision	Date	Purpose
0	27/11/2025	Document creation
1	28/11/2025	Document revision



## 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		REVIEWED AND APPROVED BY:	
Date	30/11/2025	Date	
Name	Sergio Bret Villalta	Name	
Position	Project author	Position	Project Supervisor

0. CONTENTS

0.	Contents .....	3
1.	General comments about the work progress.....	4
1.1.	Incidences .....	4
1.1.1.	WP1: .....	4
1.1.2.	WP2.1: .....	5
1.1.3.	WP4: .....	6
1.1.4.	WP2.2: .....	8
1.2.	Work Plan modifications .....	9
2.	updated work plan.....	10
2.1.	Updated Work Packages, Tasks and Milestones.....	10
2.2.	Updated Time Plan (Gantt diagram) .....	15

Document: [Project critical review_Sergi_Bret.docx]	<div>Project Critical Review</div> <div><b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b></div>	<div></div>
Date: 28/11/2025		
Rev: 01		
Page 4 of 18		

## 1. GENERAL COMMENTS ABOUT THE WORK PROGRESS

### 1.1. *Incidences*

---

#### 1.1.1. WP1:

##### MWTU Hardware Limitation (Simultaneous Read Issue):



- **Problem:** The TPU-1 equipment's web server does not support multiple simultaneous reads. When attempting to poll the alarms section of the equipment configuration webpage in a new window, the execution of other tests could not be performed.
- **Solution:** We implemented a semaphore (is\_main\_task\_running) to pause the GUI alarm monitor when we want to execute a Robot test.

##### Selenium Ghost Windows:

- **Problem:** Since the initial version of the GUI we started with, empty browser windows (with the "data:," link) have appeared in all the sessions we opened. This issue didn't affect the correct functioning of test cases, but it was annoying.
- **Solution:** We used a monkey patch technique in webdriver.Remote. In this way, we avoid newSession commands and we can reuse the existing session.

##### Architecture Error:

- **Problem:** Recurrent AttributeError due to mixing logic and UI.
- **Solution:** We made a complete refactoring, moving from having a single file to multiple files for each section/function, organized in pairs within the "gui" (where widgets and visual elements are configured) and "controller" (where the app logic is located) folders.  
app.py acts as the brain for any operation where gui\_data <-> controllers need to interact.

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 5 of 18		



### 1.1.2. WP2.1:

#### GPIO Library Incompatibility with RPi OS (Debian 13):

- **Problem:** "Failed to add edge detection" error. We were unable to configure the pins mapped in the custom server we implemented running on the RPi. The traditional RPi.GPIO library is incompatible with the new Debian 13 software that is installed within the Raspberry board.
- **Solution:** We migrated and refactored the server code (hil\_server.py) to use libgpiod.

#### Hardware Error: Reading Outputs via GPIO using TE291 Converter:

- **Problem:** We attempted to use logic level converters to read the TPU output signals. However, we gathered static readings and failed to detect level changes when an output was activated.
- **Cause:**
  - o These converters are designed for the I2C protocol, where transitions between '1' and '0' occur when a device pulls the bus to ground, producing a '0'. Since we were constantly driving a '1' (5V), no voltage change was detected.
  - o It would have worked if we had connected GND to one of the IPTU terminals instead of 5V (connecting the other terminal to the converter's HVx). However, since Raspberry Pi has internal configurable pull-up resistors that work similarly, we decided to bypass the logic converters.
- **Solution:**
  - o GPIO pins are configured in input pull-up mode to capture the T5 timestamps when the outputs are generated.
  - o We bypassed the voltage converters. Instead of routing voltage through the IPTU relays, we route GND. Upon output generation, the relay closes, pulling the configured pull-up input to ground. This voltage drop is interpreted as the output activation.

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 6 of 18		

### 1.1.3. WP4:

#### TPU Equipment Synchronization Issues:



**Context:** We implemented the complete system to execute performance tests, capable of displaying delays in different parts of the transmission during a pulse iteration (Input generation) in an End-to-End scenario. This involved two TPU equipment connected via an RJ45 cable using ICPT digital communication modules. In order to obtain the delays, we need the timestamps for the following events:

- **T0:** Instant when the relay actuation signal is generated via the feedback channel, using optocouplers, that follow the 48V output of the input activation relays (IPTUs).
  - o **Note:** For the moment, we cannot obtain this as we are waiting the arrival of the RPi-controlled optocoupled relay module. Instead, we used an Arduino module (previously developed by the department) to generate the activation signals in the IPTUs. It is limited to a single channel, and we lack the feedback signal T0. T1 (provided by the equipment via SNMP) was taken as a substitute instant instead, gathering approx. 700us hardware delay of (T1- T0), that we cannot currently account.
- **T1, INP (TPU A):** Instant when the input activation interrupt arrives at the IPTU module.
- **T2, TX (TPU A):** Instant when the MWTU processing module has processed the input and sent the command via the assigned communication module (ICPT for these tests).
- **T3, RX (TPU B):** Instant when the trip command is received at the ICPT communication module.
- **T4, OUTP (TPU B):** Instant when the MWTU module has processed the trip command and the Output is generated via the IPTU module.
- **T5:** Arrival of the Output at the Raspberry Pi via a GPIO PIN in Pull-Up mode.



*\* T0 and T5 are obtained from the logs generated by the RPi itself.*

***\* T1, T2, T3 y T4 are extracted from the SNMP traps captured by each unit. These timestamps depend on the clock configured on each device. In our case, we have both TPUs and the RPi synchronized with the department's internal NTP server clock.***

**Incident:** When the setup was ready for the performance testing, the TPU-1 equipment clocks didn't synchronize well, despite using the same local ZIV server clock. We discovered it was a hardware issue in one of the TPUs. By mixing one specific MWTU version (processing module that houses the TPU CPU) with a different version (updated with the KWTU submodule), one of them maintained a 5ms delay that skewed all results.

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 7 of 18		

**Solution:** Once the same type of MWTU was installed in both units and their firmwares were updated to the latest versions, we achieved a correct synchronization of the SNMP trap timestamps (providing the exact instant of each interruption) from each device.

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 8 of 18		

#### 1.1.4. WP2.2:

##### Command Output Reading Issue (T5 timestamp):



###### Incident:

While executing the HIL performance tests (once the equipment synchronization issue was solved), I noticed that the (T5 – T4) delays for many of the pulses in each burst test displayed highly disparate values, with abrupt variations (which didn't appear to be due a clock desynchronization) and were also non-deterministic.

This led me to conclude that we were facing a massive electrical noise problem at the Raspberry Pi input (T5).

###### Situation Analysis:

- **Impedance Mismatch and Leakage Current:** The TPU-1 output modules (IPTUs) use Solid State Relays (SSR). According to what I discovered in the TPU manual, these SSRs can have a leakage current of up to 300 uA.
- **Signal Saturation:** The Raspberry Pi's internal pull-up resistors (~50 kΩ) were insufficient to drain this leakage current. According to Ohm's Law, this resulted in a maximum (theoretically) voltage drop of aprox. 15 V, keeping the logic level in an indeterminate state.
- **Noise Susceptibility:** This high-impedance floating state caused the RPi to detect massive electromagnetic noise, registering over 4.6 million false events in seconds, making the testbench unusable for precise timing measurements.
- **Solution:** The use of a common ground and a direct electrical connection between the IPTU and the RPi was finally discarded. We are currently awaiting the arrival of the optocoupler modules. This will provide isolation and a low-impedance current sink (LED have > 5mA activation threshold) to filter the leakage and stabilize the readings.

Document: [Project critical review_Sergi_Bret.docx]	<div>Project Critical Review</div> <div><b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b></div>	<div></div>
Date: 28/11/2025		
Rev: 01		
Page 9 of 18		



## 1.2. *Work Plan modifications*

---

We have modified the original work plan due to two main factors:

1. **Hardware Delays (WP2):** The diagnosis of the leakage issue and the need for the missing material to arrive, such as the relay modules and optocouplers, have paused the physical validation phase of the HIL system. **A delay of approximately 1 week is expected for component arrival and assembly.**
2. **New WP3 - Noise Injection:** Following the technical feedback provided, the project scope has been expanded to include a dedicated **Noise Injection and Robustness Phase**. This was a high-level goal before, but now requires a specific hardware and software development that has not been taken account in the original plan:
  - **Analog Tests:** Design of an operational amplifier-based circuit (Summer + Amplifier) for IBTU module to mix TPU signals with the white noise.
  - **Digital Tests:** Integration of an external **Netstorm** (from Albedo) equipment for Ethernet impairment testing.

**Consequently, the original "WP3 Integration" has been shifted to "WP4", and a new "WP3" focused on Noise & Robustness has been introduced.**



Document: [Project critical review_Sergi_Bret.docx]	<div>Project Critical Review</div> <div><b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b></div>	<div></div>
Date: 28/11/2025		
Rev: 01		
Page 10 of 18		

## 2. UPDATED WORK PLAN



### 2.1. Updated Work Packages, Tasks and Milestones

#### Work Packages:



<b>Project:</b> Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	<b>WP ref:</b> WP1	
<b>Major constituent: Development of the Control and Testing Software</b>	Sheet 1 of 5	
<b>Short description:</b> 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	
	Start event: 06-10-2025 End event: 31-10-2025 <b>(Completed)</b>	
<b>Internal task T1:</b> Analyze 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. <b>Internal task T2:</b> 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. <b>Internal task T3:</b> 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. <b>Internal task T4:</b> 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.	<b>Deliverables:</b> T1 ✓ T2 ✓ T3 ✓ T4 ✓	<b>Dates:</b> 13-10-2025 21-10-2025 24-10-2025 30-10-2025

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 11 of 18		



<b>Project:</b> Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	<b>WP ref:</b> WP2	
<b>Major constituent:</b> Design and Construction of the HIL Test Bench (UPDATED)	Sheet 2 of 5	
<b>Short description:</b> 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: 22/12/2025	
	Start event: 31/10/2025 End event: In Progress	
<b>Internal task T1:</b> 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. <b>Internal task T2:</b> 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. <b>Internal task T3:</b> 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. <b>Internal task T4 (Update):</b> Assembly of the Optocoupler interface and wiring upon arrival. <b>Internal task T5 (Update):</b> Migration of the HIL Server code from RPi.GPIO to libgpiod to support Debian 13 interrupts correctly (as we discovered during the software tests). <b>Internal task T6:</b> 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. <i>The operation of the implemented HIL system has been verified using an Arduino module to generate triggers and activate inputs via hardware (pending to arrival the optocoupler modules to replicate this with the relay module). The system has been prepared and adapted in anticipation of using the Raspberry Pi itself for input generation (modifying a constant in the code only required).</i> <i>Performance burst tests have already been successfully done within this provisional scenario. Irregular results were only observed during the mentioned before incident caused by the leakage current in the IPTU communication module. All equipment clocks are synchronized via NTP with the local ZIV server, achieving an accuracy of approximately 1ms.</i>	<b>Deliverables:</b> T1 ✓ T2 ✓ T3 ✓ T4 T5 ✓ T6	<b>Dates:</b> <b>T4:</b> 05/12/25 <i>(We are awaiting the arrival of the remaining components to assemble the final system and execute the performance tests)</i>  <b>T6:</b> 22/12/25

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 12 of 18		

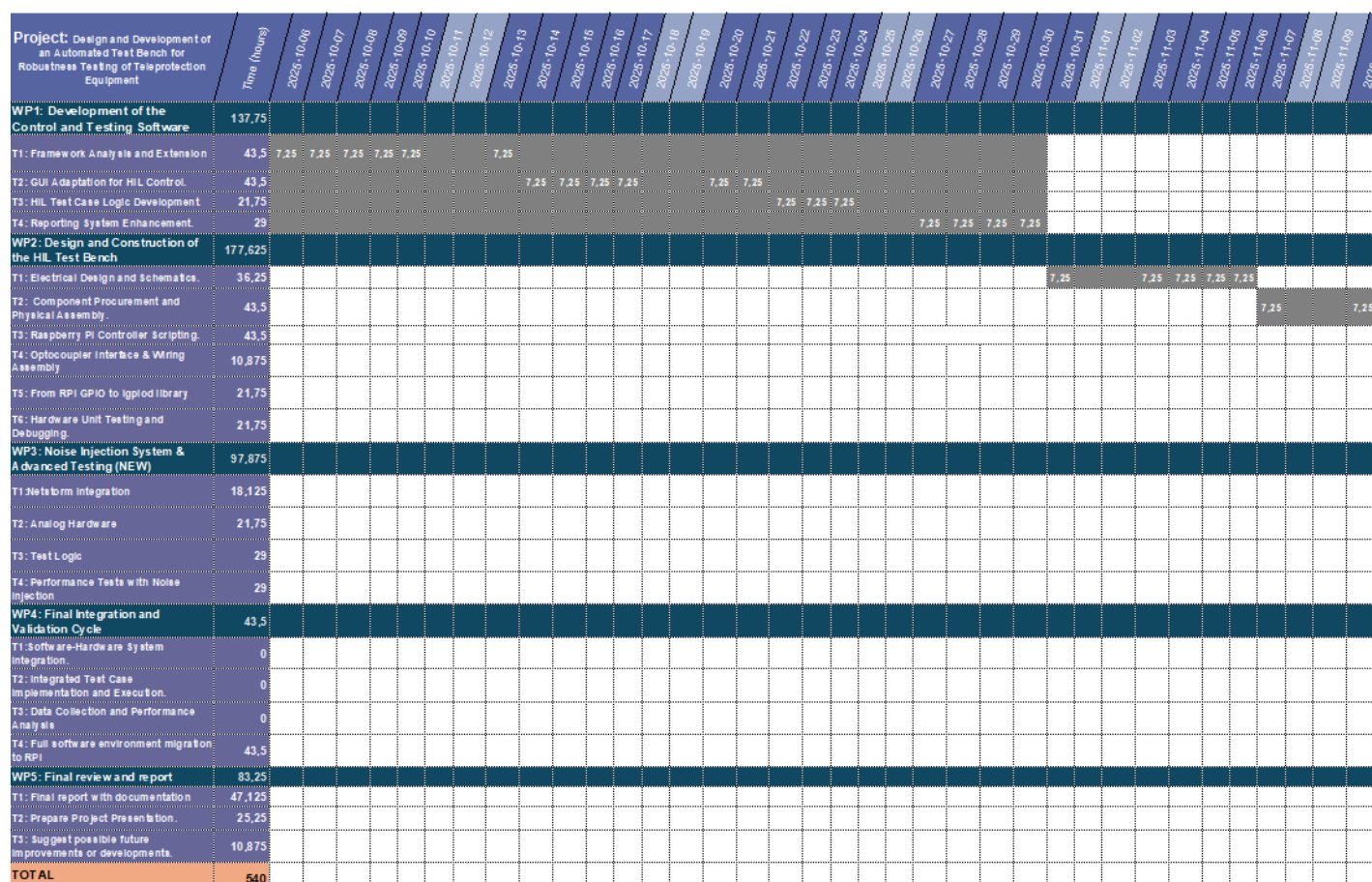
<b>Project:</b> Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	<b>WP ref:</b> WP3	
<b>Major constituent:</b> Noise Injection System & Advanced Testing ( <b>NEW</b> )	Sheet 3 of 5	
Short description: Design of analog signal conditioning hardware and integration of digital network impairment tools to perform Obedience and Security testing (IEC 60834-1).	Planned start date: 28/11/2025	
	Planned end date: 24-12-2025	
	Start event: 28/11/2025 End event: <a href="#">In progress</a>	
<b>Internal task T1 (Netstorm Integration):</b> Research and implementation of a control library for the Albedo Netstorm equipment (investigating TCP/API capabilities vs VNC automation) to simulate packet loss and jitter.  <b>Internal task T2: (Analog Hardware):</b> Design and testing of a generic OpAmp-based circuit (Summer and Non-inverting Amplifier) to mix TPU command signals with White Noise.  <b>Internal task T3 (Test Logic):</b> Development of new Robot Framework keywords for "Obedience" (Signal + Noise) and "Security" (only Noise) tests.  <b>Internal task T4:</b> Execute performance tests at different noise levels using both the Netstorm integration and the developed analog hardware.	<b>Deliverables:</b> T1 T2 T3 T4	<b>Dates:</b> 03-12-2025 10-12-2025 16-12-2025 24-12-2025

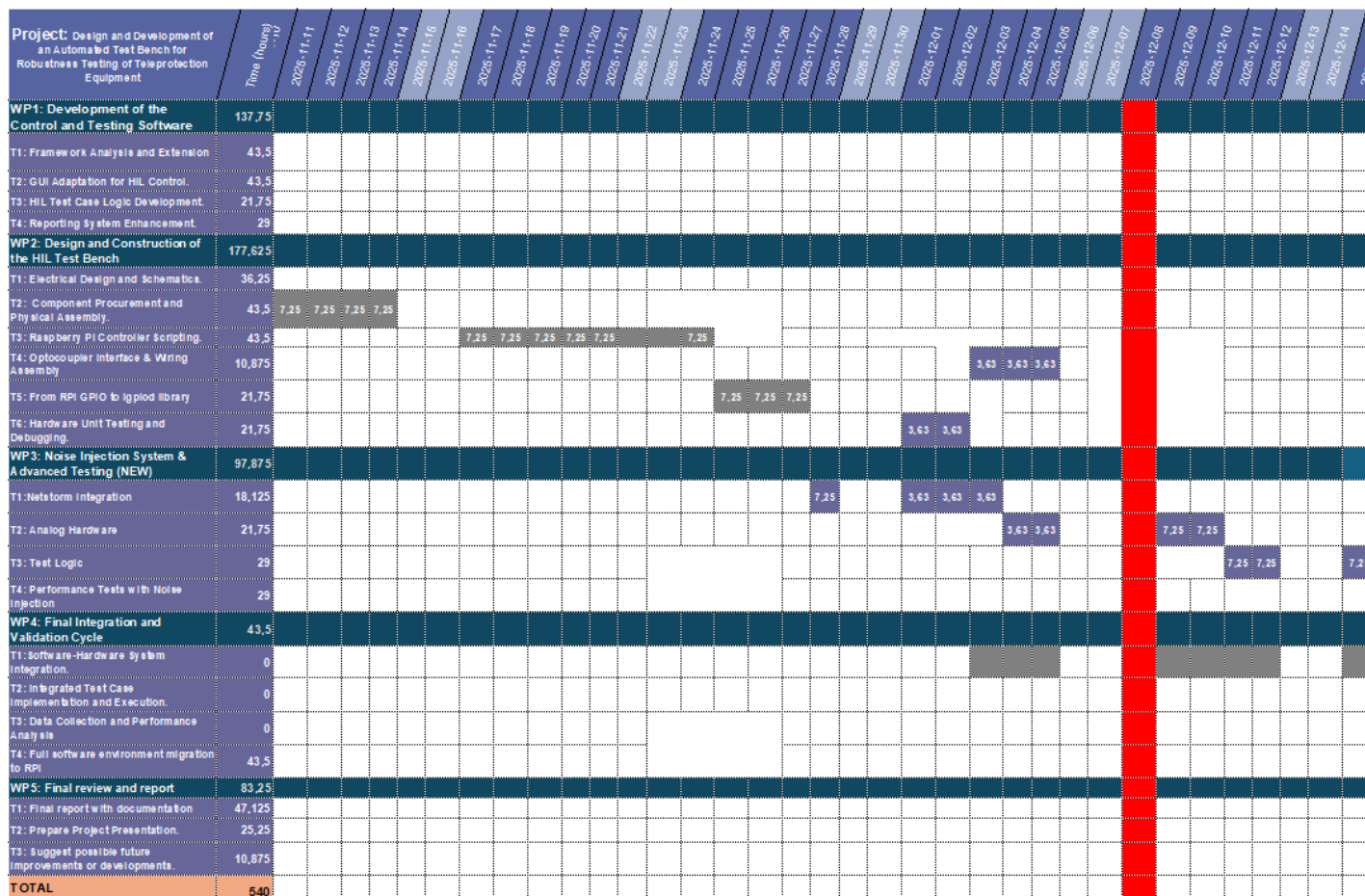
Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 13 of 18		

<b>Project:</b> Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	<b>WP ref:</b> WP4	
<b>Major constituent:</b> <b>Final Integration and Validation Cycle (SHIFTED)</b>	Sheet 4 of 5	
<b>Short description:</b> In this 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.	Planned start date: 03/12/2025 Planned end date: 05/01/2026	
	Start event: 04/11/2025 End event: In progress	
<b>Internal task T1:</b> 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. <b>Internal task T2:</b> 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. <b>Internal task T3:</b> Collect the automated reports and logs generated by Robot Framework and TPU-1's chronological log. Analyze the performance data, focusing on command execution times, system reliability, and identifying any bugs or operational limits. <b>Internal task T4 (Optional):</b> 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.	<b>Deliverables:</b> T1 ✓ T2 ✓ T3 ✓ T4 -> We'll <b>prioritize completing the final HIL System testing</b> (once the components arrive) and carrying out WP3 (Noise Injection Noise).	<b>Dates:</b> <b>T4:</b> 05/01/2025 (If we have enough time)

Document: [Project critical review_Sergi_Bret.docx]	<b>Project Critical Review</b> <b>[Design and Development of an Automated TestBench for Robustness Testing of Teleprotection Equipment in Electrical Substations]</b>	 
Date: 28/11/2025		
Rev: 01		
Page 14 of 18		

<b>Project:</b> Design and Development of an Automated Test Bench for Robustness Testing of Teleprotection Equipment	<b>WP ref:</b> WP5	
<b>Major constituent: Final review and report</b>	Sheet 5 of 5	
<b>Short description:</b> 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: <a href="#">07/01/2026</a> Planned end date: <a href="#">23/01/2026</a>	
	Start event: End event:	
<b>Internal task T1:</b> Prepare a final project report with documentation of the entire project, including the final design of the testbench, the implemented software, the test results, and the analysis. Write the final TFG memory.	<b>Deliverables:</b> T1 T2 T3	<b>Dates:</b> <a href="#">16-01-2026</a>
<b>Internal task T2:</b> Prepare Project Presentation.		<a href="#">09-01-2026</a>
<b>Internal task T3:</b> Suggest possible future improvements or developments.		<a href="#">23-01-2026</a>





[illegible]

[illegible]