# Test-Driven Development (TDD) Style: QT Display Test Cases

## Overview

The following test cases follow a Test-Driven Development process for validating the Qwt-based QT interface, which includes a thermo widget, curve plot, and table. Each test begins with defining the requirement and expected behavior, followed by implementation and verification.

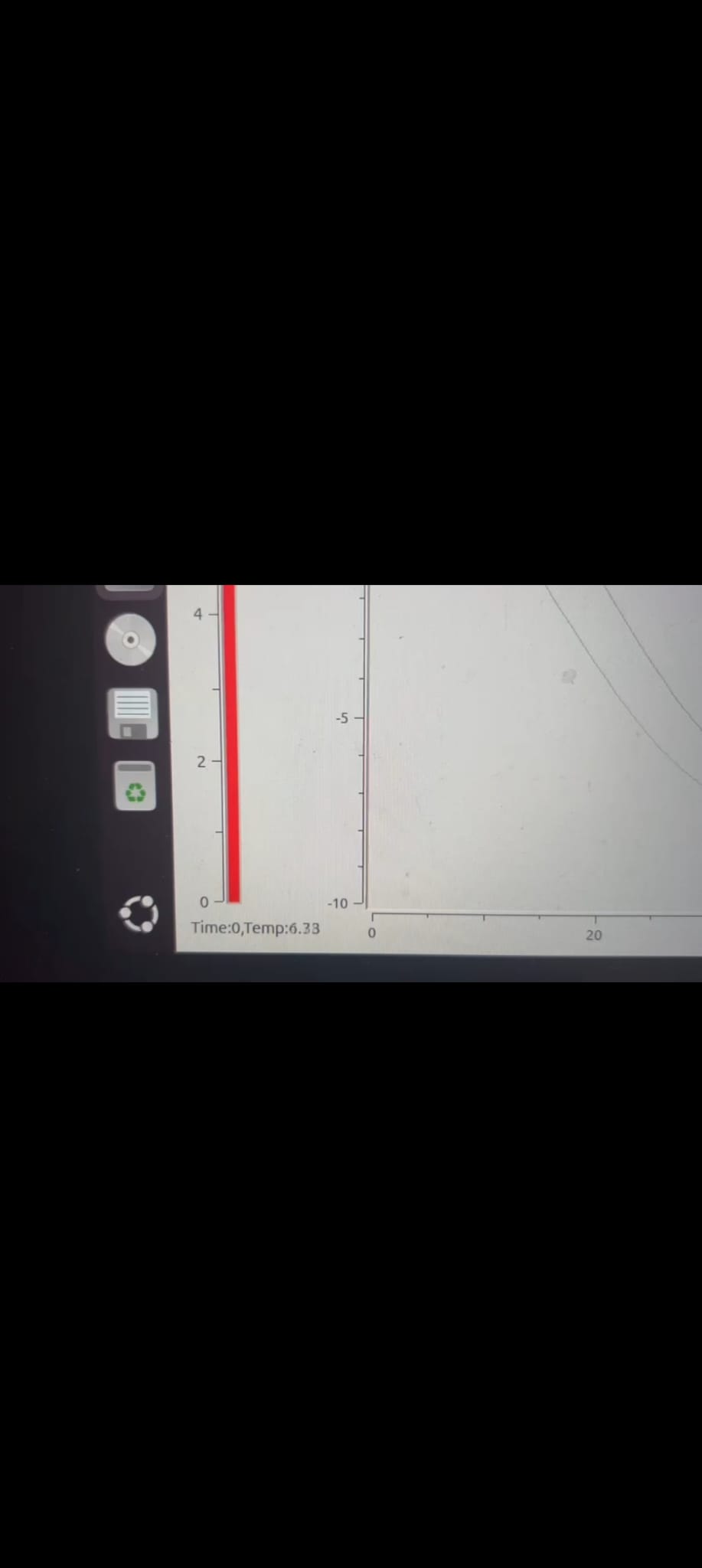
## QT-01: Display Static Thermo

Test Goal: Verify that Qwt Thermo launches and displays a predefined constant value.

Expected Behavior:  
Qwt Thermo appears on screen  
Displays the constant temperature value

Implementation Notes:  
Launch sample Thermo program

Test Result: Passed



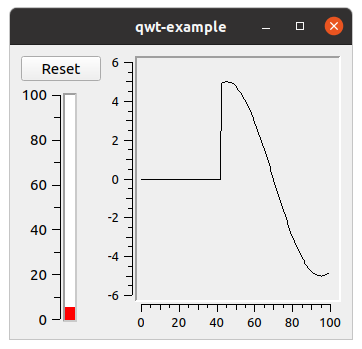
## QT-02: Timer-Based Curve Plot

Test Goal: Verify that a Qwt curve updates periodically using a timer to simulate dynamic plotting.

Expected Behavior:  
Curve updates automatically  
X-axis reflects time, Y-axis reflects temperature

Implementation Notes:  
Use CppTimer class to feed fake temperature data to QwtPlot

Test Result: Passed



## QT-03: Combined Layout – Thermo, Curve, Table

Test Goal: Ensure that Qwt Thermo, Plot, and Table are visible in a unified, horizontal layout.

Expected Behavior:  
All widgets visible without overlap  
Proper spacing and alignment

Implementation Notes:  
Header in Qwt Table was initially cut off

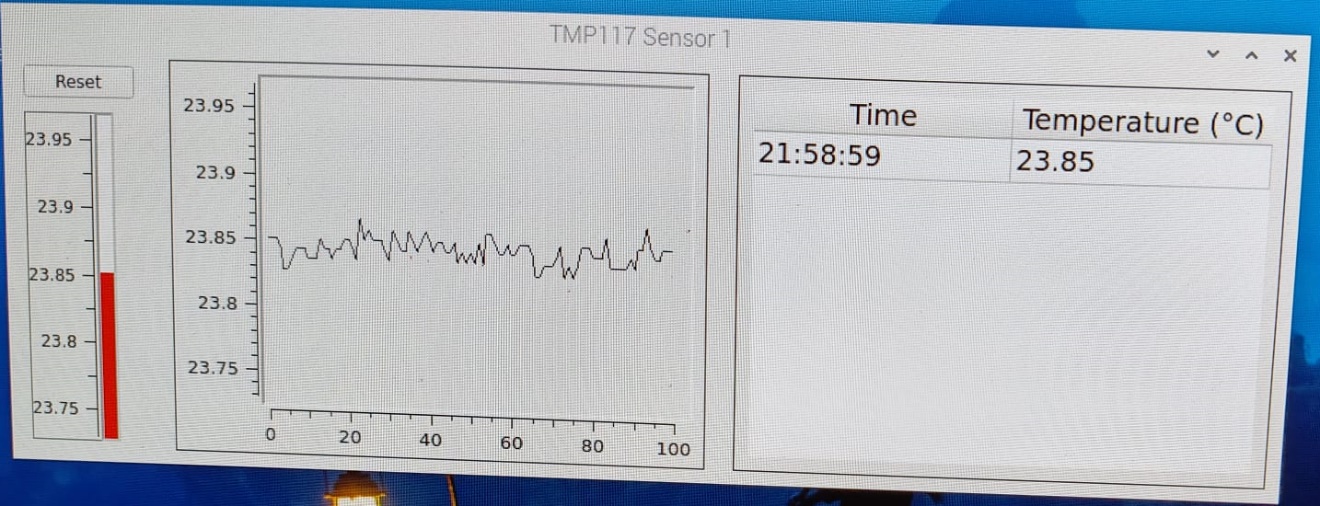
Test Result: Passed with UI improvement noted

## QT-04: Qwt Table — No Cutoff

Test Goal: Ensure the Qwt Table displays DateTime and Temperature values fully.

Expected Behavior:  
No clipped text in headers or cells

**Test Result: Passed**



## QT-05: Thermo Value Accuracy

Test Goal: Confirm Qwt Thermo displays real-time temperature values accurately.

Expected Behavior:  
Real-time sensor data shown in widget  
Value refreshes dynamically

Test Result: Passed

## QT-06: Dynamic Thermo Scaling

Test Goal: Test that Qwt Thermo widget automatically scales when it receives data outside its initial range.

Expected Behavior:  
widget to auto-rescale based on new min/max values to display the current temperature correctly

**Test Result: Passed**

## QT-07: Extra Feature - Threshold-Based Thermo Color

Test Goal: Check color changes in Thermo widget based on temperature thresholds.

Expected Behavior:  
Below threshold = Blue  
Normal = Green  
Above = Red

Implementation Notes:  
Simulate data using heat/cold sources

Test Result: Not run.

## QT-08: Real-Time Curve Accuracy

Test Goal: Ensure plotted curve reflects actual temperature data dynamically.

Expected Behavior:  
Smooth curve updates without lag  
Accurate Y-axis scaling; recalculate as per the max and min temperature value in the plot array to display all the values.

**Test Result: Passed**

QT-09: Extra Feature - Threshold Based Curve Color

Test Goal: Check color changes in plotted curve based on temperature thresholds.

Expected Behavior:

Below threshold = Blue

Normal = Green

Above = Red

Implementation Notes:

Simulate data using heat/cold sources

Test Result: not Run

QT-10: Table Value Accuracy

Test Goal: Validate that the table shows accurate DateTime and temperature logs.

Expected Behavior:  
Table display the correct timestamp and temperature values of last reading.

**Test Result: Passed**

QT-11: Extra Feature :Threshold Based table value Color

Test Goal: Check color changes in plotted curve based on temperature thresholds.

Expected Behavior:

Below threshold = Blue

Normal = Green

Above = Red

Implementation Notes:

Simulate data using heat/cold sources

Test Result: not Run

QT-12: Sensor 2 – Repeated Tests

Test Goal: Repeat all above tests using a second sensor.

Expected Behavior: execute all the above test cases.

**Test Result: all basic test case passed Extra Feature Not Run (under development)**

