

## 6. SUMMARY

We have looked at what a tachograph is and what regulations dictate the use of tachographs. According to the EU legislation, tachographs are devices that measure the speed, travelled distance and driver activity in a vehicle and record that data. The law specifies that such a device must be installed in any commercial vehicle above a certain mass and above a certain number of people that can fit in it. Aside from that, tachographs can be connected to other devices in a vehicle such as a dashboard, vehicle position antenna, etc.

Before being put to use a tachograph must be programmed and calibrated, meaning that the tachograph must have the necessary parameters for interoperability with the connected hardware as well as be programmed to measure accurate vehicle speed, travelled distance and driver activity. To calibrate a vehicle with a tachograph the vehicle must be taken to a workshop where various tests will be done to program the tachograph with the necessary parameters. In a workshop, a vehicle is tested using equipment such as Rolling Roads and Flexi-switches that output data collected from a driven vehicle to a tachograph programming device that uses it to calculate the right values and write the calibration factors to the tachograph.

One such tachograph programming solution is the Optimo2 tool designed and manufactured by Stoneridge Electronics Ltd. which consists of a Windows tablet connected to an Interface Board that is in turn connected to the workshop devices used for measurements and to the tachograph either by wire or wirelessly. Optimo2 comes installed with custom application software, written using .NET and WPF frameworks and the C# programming language, developed by Stoneridge that allows Optimo2 to program and calibrate tachographs using equipment available in a workshop. The author of this work is part of the development team for the Optimo2 application software where he acts as a System Verification Engineer tasked with testing the software and creating automated test setups and writing test scripts to ensure a functional high-quality software is delivered to the customer.

The testing system for the Optimo2 software is using the NUnit testing framework for C# for writing and running the automated tests. The tests must simulate real-world use scenarios for the software, for that PWM signal generators are used to simulate signals that both the tachograph and Optimo2 should receive when doing a calibration. To control the tablet VNC software is used, which allows screen capture and remote control of Optimo2 to run tests from a computer connected to it. To assert the test flow the test system uses a set of images that it expects to see in certain areas of the screen at certain steps during the test execution and a code library that compares these images to the screen captured by VNC in real-time.

This image comparison method creates several problems. The fact that images are used for comparison means that those images must be prepared in advance, if any change in the UI was made (even moving a button by a few pixels) during development it means that the automated tests will fail until the images used are updated. If the images used contain text they can only be used for testing the software in one language and must be recreated for testing other languages. Optimo2 is available with 2 tablet options and has 2 different screen resolutions, requiring that separate images are made for each tablet. All of these factors make maintenance very complicated and writing new tests very time-consuming.

This work aims to see if UI Automation can be used as an alternative to the VNC and image comparison for controlling the tested software on the tablet. UI Automation is an accessibility framework made by Microsoft to programmatically control Windows applications that are written using the WPF, Windows Forms, UWP, WinUI or MAUI frameworks. It allows finding and controlling UI elements in an application because they are visible to the OS itself. After analyzing possible solutions like TestStack.White, FlaUI, Winium, and WinAppDriver + Selenium it was decided to use the last option. Selenium used with WinAppDriver allows automating UI on remote devices and is the method suggested by Microsoft themselves for testing Windows applications.

Several test scripts were written for Optimo2 software using this solution, automating 3 scenarios that were already available in the current test system. The ease of writing the tests was noted compared to the previous solution. The tests written with UI Automation also make use of the same resource translation tool as the tested software, making these tests available in any supported language. They also proved to be around 15% faster to execute compared to their predecessors. A test that was new to both the new and old solutions was created to compare the time needed to create tests, 1.5 hours for the new solution (compatible with all languages and screen resolutions), against the 2 hours used for the same test using the old solution plus around 1 hour for each language and tablet resolution that is to be added. The absence of images also frees the amount of storage space occupied by the solution, decreasing its size by up to 500 MB.

This modification to the testing process of the Optimo2 software allows for better, more detailed testing, faster testing times, better test coverage, and faster software development cycle iterations and is an overall improvement over the previous solution.