

Purpose:

The goal of this project is to implement a software tool to help our test team to test our 4D radar product. After testing the radars, our test team can equip the radars into our test cars for real road testing. Also, They can make plans based on the testing time and the total number of radars our customer required.

This project provides the features below.

- 1) Control the test machine in our lab to test our 4D radars.
- 2) Provide the UI interface to our test team. Our test team can use it to configure test parameters for the testing.

Key considerations:

- 1) Control the test machine to move our 4D radar vertically and horizontally.
- 2) After moving the 4D radar to turn 1 degree angle, it collects UDP data packages from the radar via the network.
- 3) After collecting the UDP data packages, it writes the raw data into files. So our hardware engineers and radar engineers can use these files to calibrate the radar.
- 4) Provide the UI interface for our test team to start or stop the test machine. Also, they can control the moving speed of the test machine.
- 5) Provide the UI interface for our test team to move the test machine to a specific angle.
- 6) Provide the UI interface for our test team to do different types of testing. They can choose to do a quick test or a full test.
- 7) Generate different types of output files based on their configurations.
- 8) Provide the UI interface for our test team to configure the input parameters for the testing such as the starting position, horizontal moving range, vertical moving range and so on.
- 9) Provide the UI interface for our test team to configure the output parameters for the testing such as output files saved path, the name format of output files and so on.

Implementations:

- 1) Using Qt GUI to provide the UI interface based on the requirements.
- 2) Using Qt Network to collect the UDP data packages from the test radars.
- 3) Using Qt File Systems to save data into files.
- 4) Using Qt Thread to control the test machine.
- 5) Using Qt Multithreading and Qt Signal-slots mechanism to coordinate the thread of controlling the test machine, the thread of writing data into files and the thread of collecting UDP data packages.
- 6) Using OOP Design Principles to implement different classes that take single responsibility for their single task. Moving these classes into different threads to collaborate to do the entire work by using the Qt Signal-slots mechanism.

Value:

- 1) In the beginning, our test team only test the 4D radars in our test cars. Usually, some radars don't work well even if they are manufactured in the same batch. In this situation, it makes their test work more difficult and wastes them a lot of time. By using this application, they can test the radars before real road testing. Thus, this project saves their time and more easier to make plans.
- 2) Our test team can use this software to do different types of testing.
- 3) Our hardware engineers and radar engineers can use this software to calibrate the radars in their work. They can quickly get the test results so that they can quickly improve their work.
- 4) Our company can know the problems why some radars don't test well based on the test results. Thus, they can improve the manufacturing process.
- 5) Our company can make plans for testing after doing a full test on a radar with this software.