

# VarastoRobo – QtClient

Jarno Poikonen, Eemeli Tikka TVT17SPL  
Oulu University of Applied Sciences  
Information Technology, Option of Device and Product Design

## Introduction

The aim of this project was to make an autonomous storage management system, in which multiple devices work in co-operation to transport ordered packages to a drop-off point. In addition a drone was to function as surveillance system for the premises.

## Methods

The user interface was implemented using Qt Framework 5.13. The GUI consists of labels, spinboxes, checkboxes, textedit elements and one graphics view element in the middle of the screen that were all customized using the design mode of the Qt IDE.

## Results

All critical features demanded from the QtClient were successfully implemented. These were showing VarastoRobo live map parsed from broadcasted UDP datagrams, emergency stop, product ordering to fixed destination coordinates, remote control (moving), capability to send, receive, handle and act on various messages defined by the mutual communication specification.

Implementation of some less critical features were unsuccessful due to time spent on fixing issues related to critical core features.

## Conclusions

Some of the Qt Framework philosophy and Object-Oriented Programming principles were ignored in order to achieve desired results in the shortest time possible.

The codebase for QtClient is currently due for some refactoring before any attempts to implement new features.

## References

- VarastoRobo GitHub:  
<https://github.com/Jarno-Poikonen/VarastoRobo>
- Qt 5.13 Documentation:  
<https://doc.qt.io/qt-5/index.html>

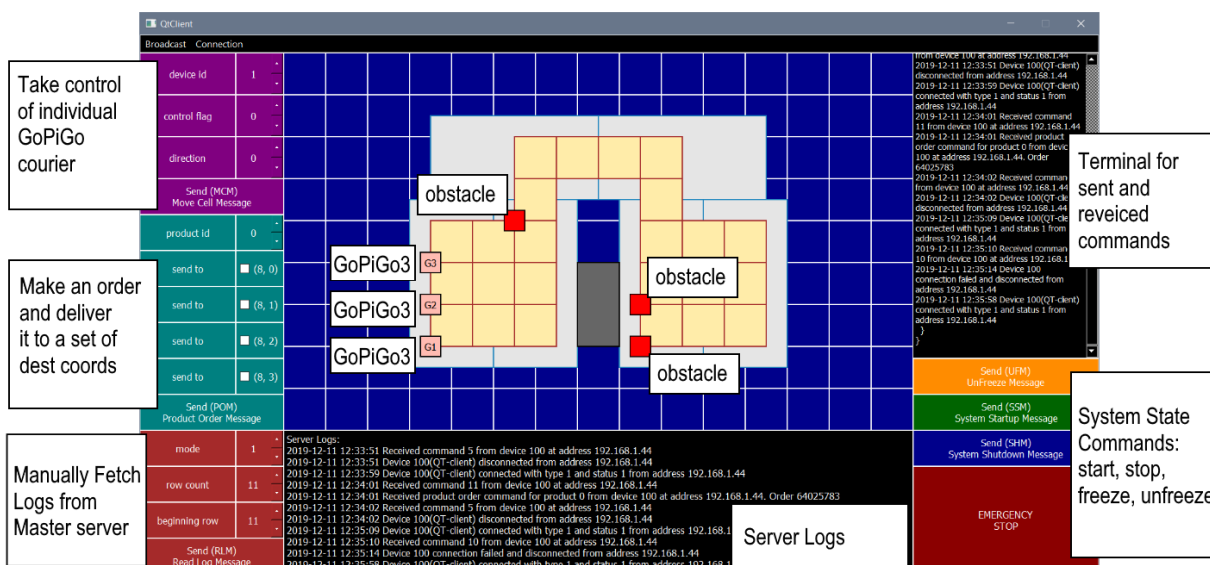


FIGURE 1. QT-Client

## Objectives

The objective was to provide the project with a UI-client with the capability to make an order, take direct control of a GoPiGo, display a real time map and enable an emergency stop for the whole system.

Any actionable events being raised from clickable elements were handled by functions defined in the custom QtClient class.

Most of the events were related to sending and receiving specific messages to the Master according to mutually agreed upon specification of the VarastoRobo system.

Communication was handled in an asynchronous fashion by utilizing the Qt Framework provided socket classes.