# Onderoek hoe werkt een (linux) queue

In dit onderzoek word er onderzocht hoe een linux queue werkt voor communicatie tussen de websocket en het RTOS. Ook word er onderzocht of er alternatieve zijn voor de linux queue.

**Het resultaat**

Uit uitvoerig onderzoek en meerdere experimenten met een linux queue vinden we dit geen goede oplossing. De oplossing is te complex om goed te implementeren naar onze mening. Als vervanging maken we de websocket nu onderdeel van het RTOS en gebruiken hier een STD::queue.

**Hoe werkt een STD::queue**

De werking van een std queue is ongeveer gelijk aan de werking van een vector. De std queue werkt via het FiFo principe. Dit betekent first in first out, dus het element dat er als eerste word ingestopt komt er ook weer als eerste uit. De std queue ondersteund de volgende commando’s:

|  |  |
| --- | --- |
| **Functie** | **Werking** |
| empty | Checkt of de queue leeg is of niet |
| size | Geeft de grootte van de queue |
| front | Geeft het eerste element uit de queue terug |
| back | Geeft het laatste element uit de queue terug |
| push(std::String) | Stopt een element in de queue |
| Pop | Verwijderd het eerste element uit de queue |
| Swap | Draait de inhoud van de queue om. |
|  |  |

**Hoe maken je een STD::queue aan**

Er kan een nieuwe std queue worden gemaakt door het volgende commando in de code te gebruiker: queue <string> commands;. Om dit te gebruiken dienen wel de libraries <string> en <queue> geinclude te worden. Tevens moet er worden gelinkt met de flag –lrt.