AstaZero AB

AstaZero - ASTA Common

|  |  |
| --- | --- |
| Performed by |  |
| \_\_Signature\_1 | \_\_Signature\_2 |
| Karl-Johan Ode |  |

Chronos

Architecture (Draft)



# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Date | Issuer | Change description |
| B01 | 2017-04-03 | Sebastian L L | First draft |

# Index

[Revision history ii](#_Toc479846610)

[Index iii](#_Toc479846611)

[1 Introduction 1](#_Toc479846612)

[1.1 Goals 1](#_Toc479846613)

[1.2 Abbreviations 1](#_Toc479846614)

[2 Architecture 2](#_Toc479846615)

[2.1 System model 2](#_Toc479846616)

[2.2 System Control 2](#_Toc479846617)

[2.2.1 User commands 3](#_Toc479846618)

[2.3 Logger 3](#_Toc479846619)

[2.4 Supervision 3](#_Toc479846620)

[2.5 Visualization 3](#_Toc479846621)

[2.6 Target Control 3](#_Toc479846622)

[2.7 Message bus 4](#_Toc479846623)

[2.8 Signals 4](#_Toc479846624)

[2.9 Source code 4](#_Toc479846625)

[3 Functions 4](#_Toc479846626)

[3.1 Synchronization points 4](#_Toc479846627)

# Introduction

## Goals

The main

## Abbreviations

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

# Architecture

In this section are fundamental quantities and equations defined. The table below lists quantities with units

## System model

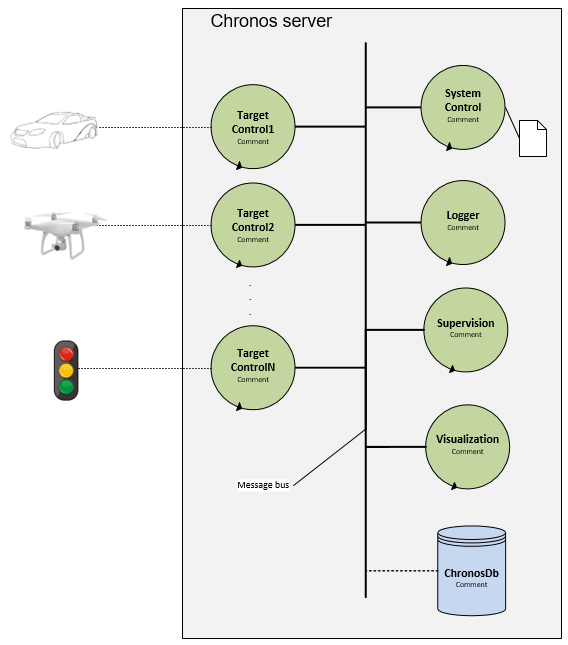


Figure 1 System model

## System Control

System control handles commands from the user.

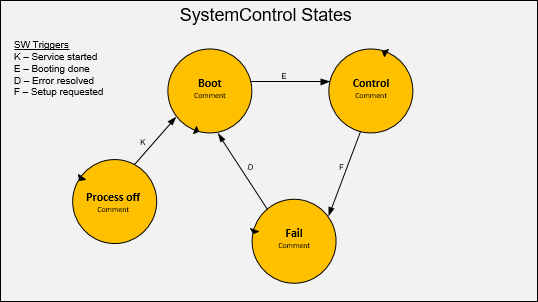


Figure 2

### User commands

|  |  |
| --- | --- |
| **Command** | **Description** |
| status |  |
| arm |  |
| start |  |
| stop |  |
| replay |  |
| control |  |
| exit |  |
| cx | Quit current command |
| cc | Check current command |

## Logger

Logger saves data going on the message bus.

## Supervision

Supervision supervises the system and handles critical situations.

## Visualization

Visualization sends data from message bus to the visualizati\*on tool.

## Target Control

Target control controls a specific object on the test area.

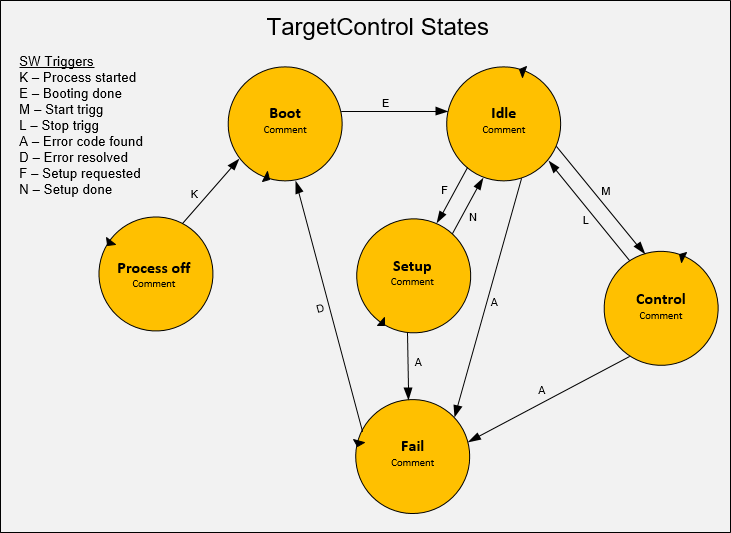


Figure 3

## Message bus

Message bus is used by the processes to communicate internally on the server.

## Signals

## Source code

## Coordinate system

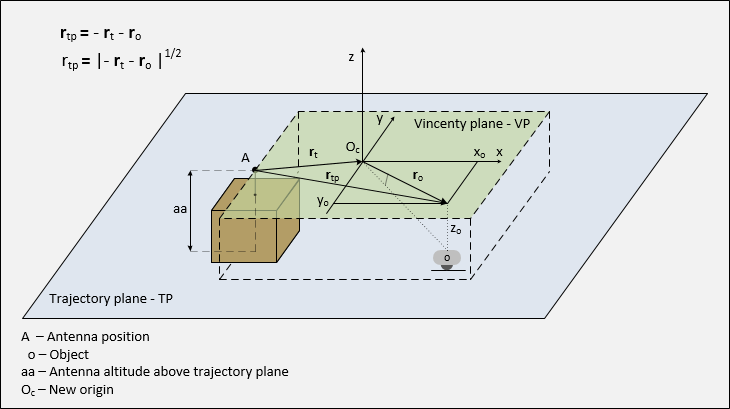


Figure 4

rt + ro1 + rtp = 0

# Functions

## Synchronization point

A synchronization point is defined by

*All objects need to reach their destination points dp at the same time as the synchronization object Os reaches the synchronization point ds*

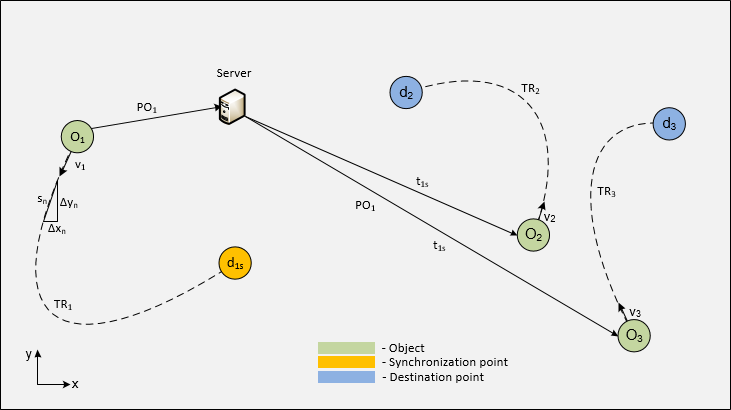


Figure 5 Synchronization point setup

Figure 2 shows a setup with three different objects O1, O2, O3. Each object has its own trajectory specified in TRn and distributed to the objects by the server. The destination point d1s is configured in the setup as the synchronization point and O1 is heading for this point. When O1 travels towards d1s it constantly sending its position and speed to the server in PO1. The server uses this information to calculate the momentaneous time t1s and sending this information to O2 and O3. t1s specifies the time until the O1 reaches d1s.

O2 and O3 uses this time information to calculate the needed speed to reach their destinations at the same time as O1 reaches d1s.

### Synchronization time

When PO1 is received by the server is the total distance s1 left to travel before reaching the synchronization point calculated by ObjectControl. This distance is divided with the current speed v1 to get the momentaneous synchronization time t1s and this value is sent to all other objects.

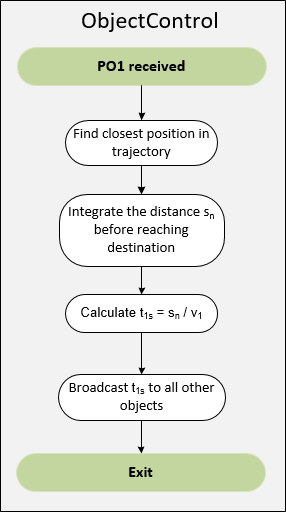


Figure 6