

## Abstract

Has been decided that Sirius will use EPICS as its distributed control system and this year the development of its High Level Applications (HLAs) started. Three development frameworks were chosen for building these applications: CS-Studio, PyQt and Matlab Middle Layer (MML). Graphical user interfaces (GUI) and machine applications have already been designed and implemented for a few systems using CS-Studio and PyQt: slow orbit feedback, lifetime calculation and top-up injection. Specific Sirius data structures were added to the MML scripts in order to allow for EPICS communication through LabCA.

## Application Environment

- **Machine Applications:** EPICS servers used to compute beam dynamics algorithms such as lifetime calculation and slow orbit feedback. Since they communicate with hardware through other EPICS servers (IOCs), they can be interpreted as EPICS clients too.
- **IOCs:** Also EPICS servers, but their main purposes are to control and establish communication with hardware such as power supplies and BPMs.
- **Client Applications:** EPICS clients built to control and monitor PVs.

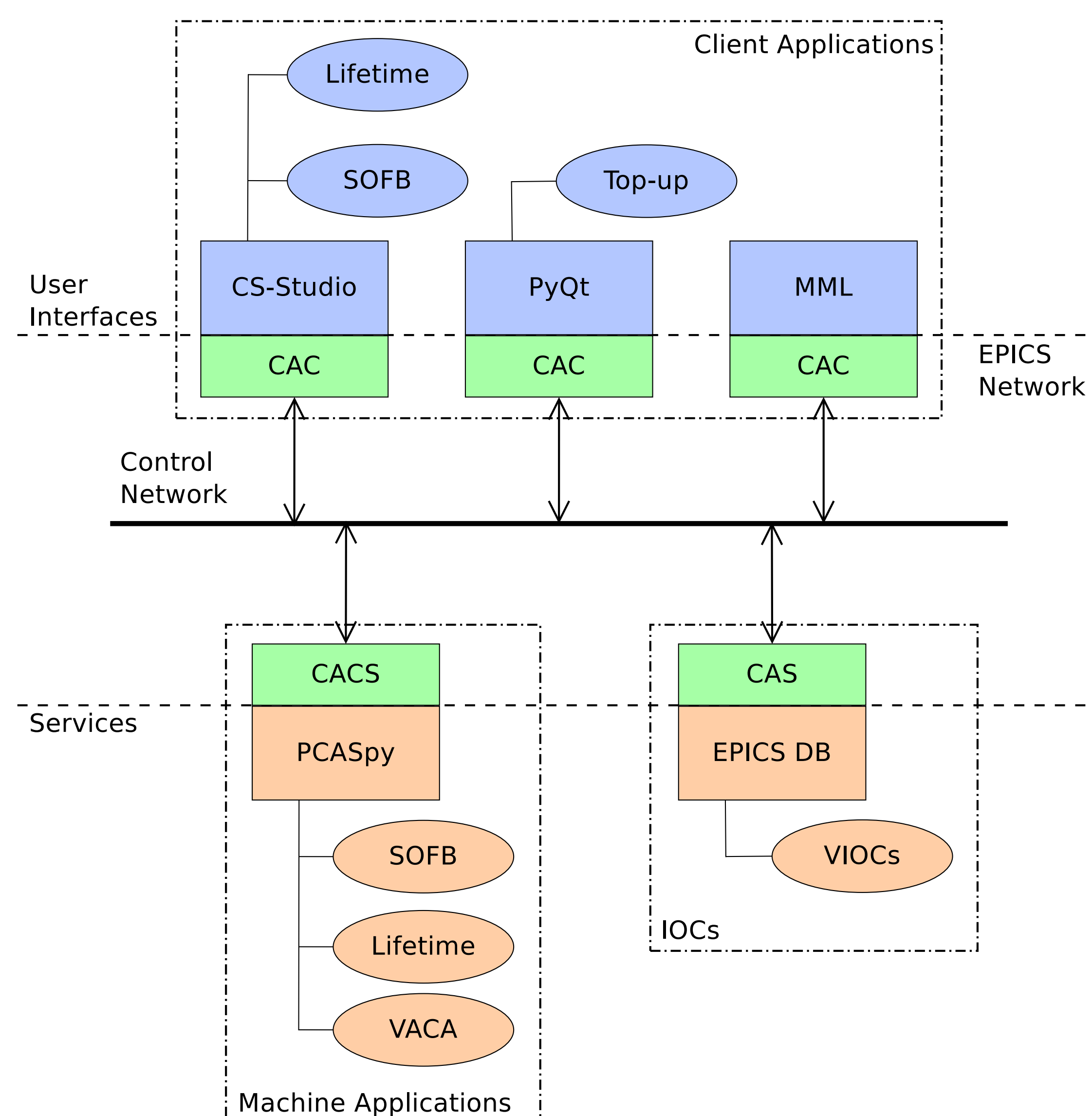


Figure 1: Application environment schematic.

## Slow Orbit Feedback

### Implemented functionalities

- **SOFB Machine Application:** Orbit/response matrix measurements and orbit corrections based on singular value decomposition (SVD), device selection, response matrix and reference orbit configurations, variable-size buffers with orbit data for averages, optional inclusion of RF frequency in the correction loop and corrector strength adjustments.
- **SOFB Client Application:** Plot displayed with measured orbit, widgets set for manual correction, the machine application mode selection and widgets for configuring sampling parameters for orbit average calculations.

### Orbit Correction

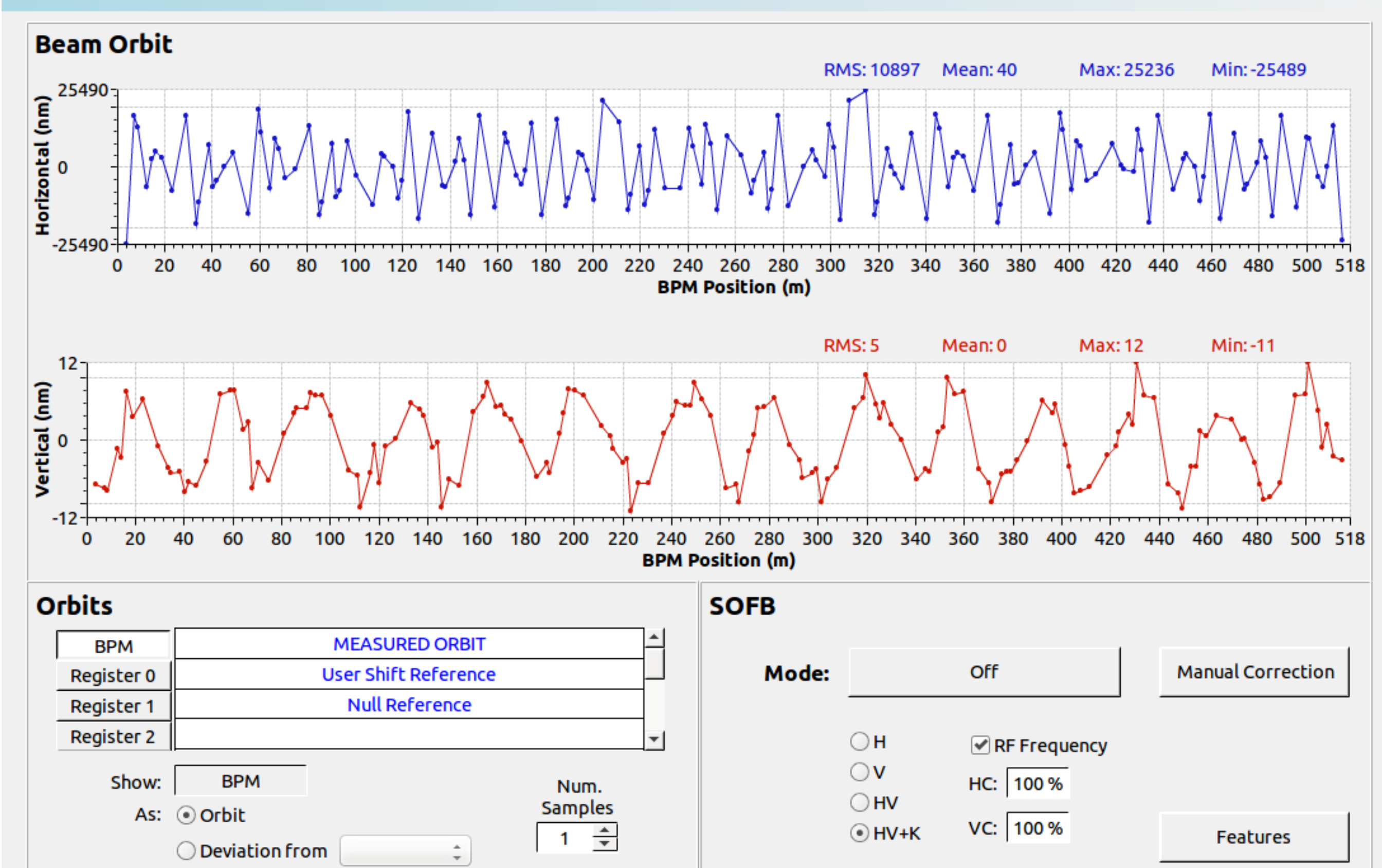


Figure 3: Orbit correction display application in the CS-Studio environment.

## Top-up Injection

### Implemented functionalities

- **Top-up Client Application:** Current and filling pattern display, injection mode selection, maximum current and maximum current decay adjustment.

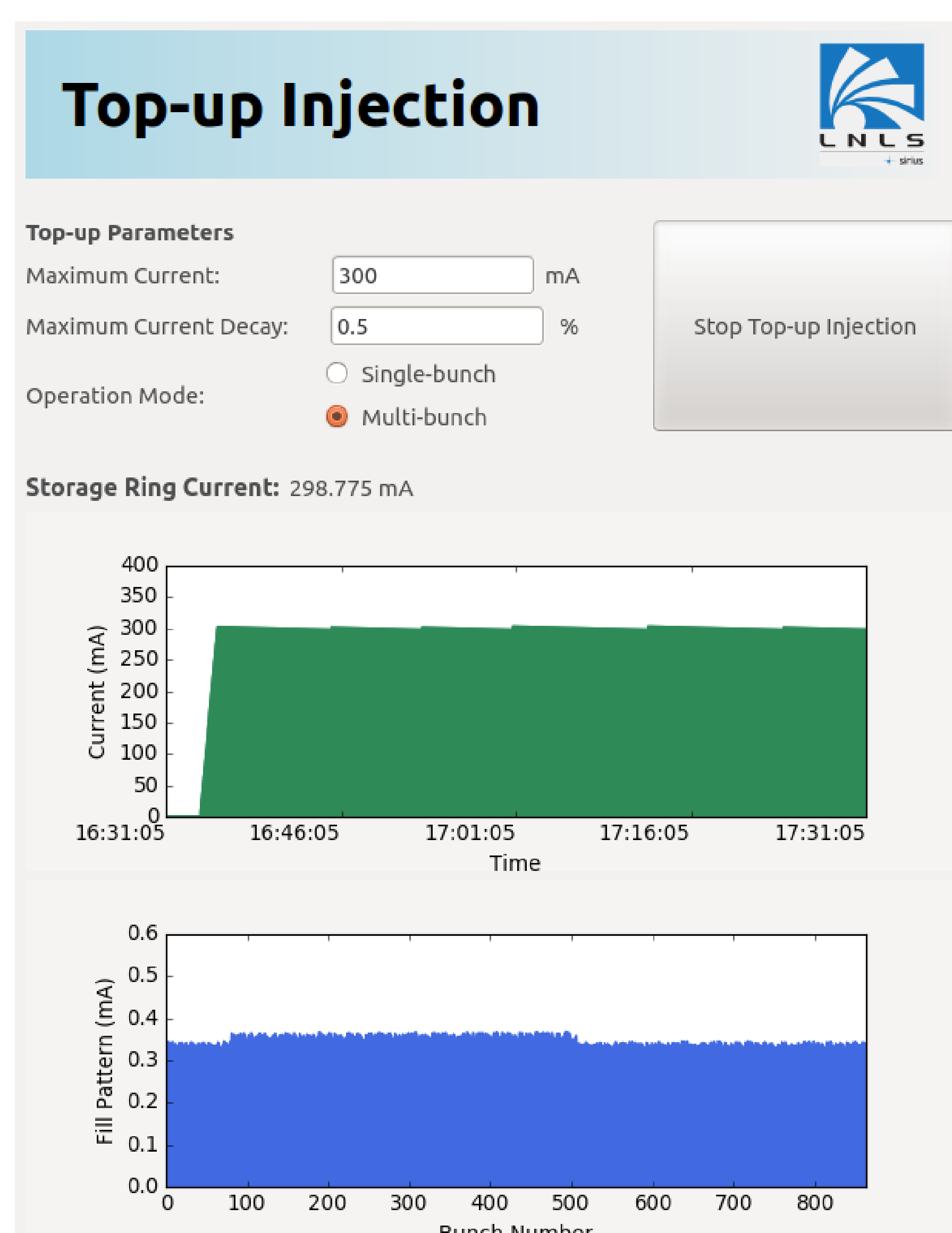


Figure 2: Top-up injection display application developed with PyQt.

## Lifetime Calculation

### Implemented functionalities

- **Lifetime Machine Application:** Lifetime calculation and precision and sampling time input parameters.
- **Lifetime Client Application:** Current and lifetime display, lifetime unit selection, lifetime graphical display, algorithm parameters display and precision and sampling time adjustment.

### Beam Lifetime

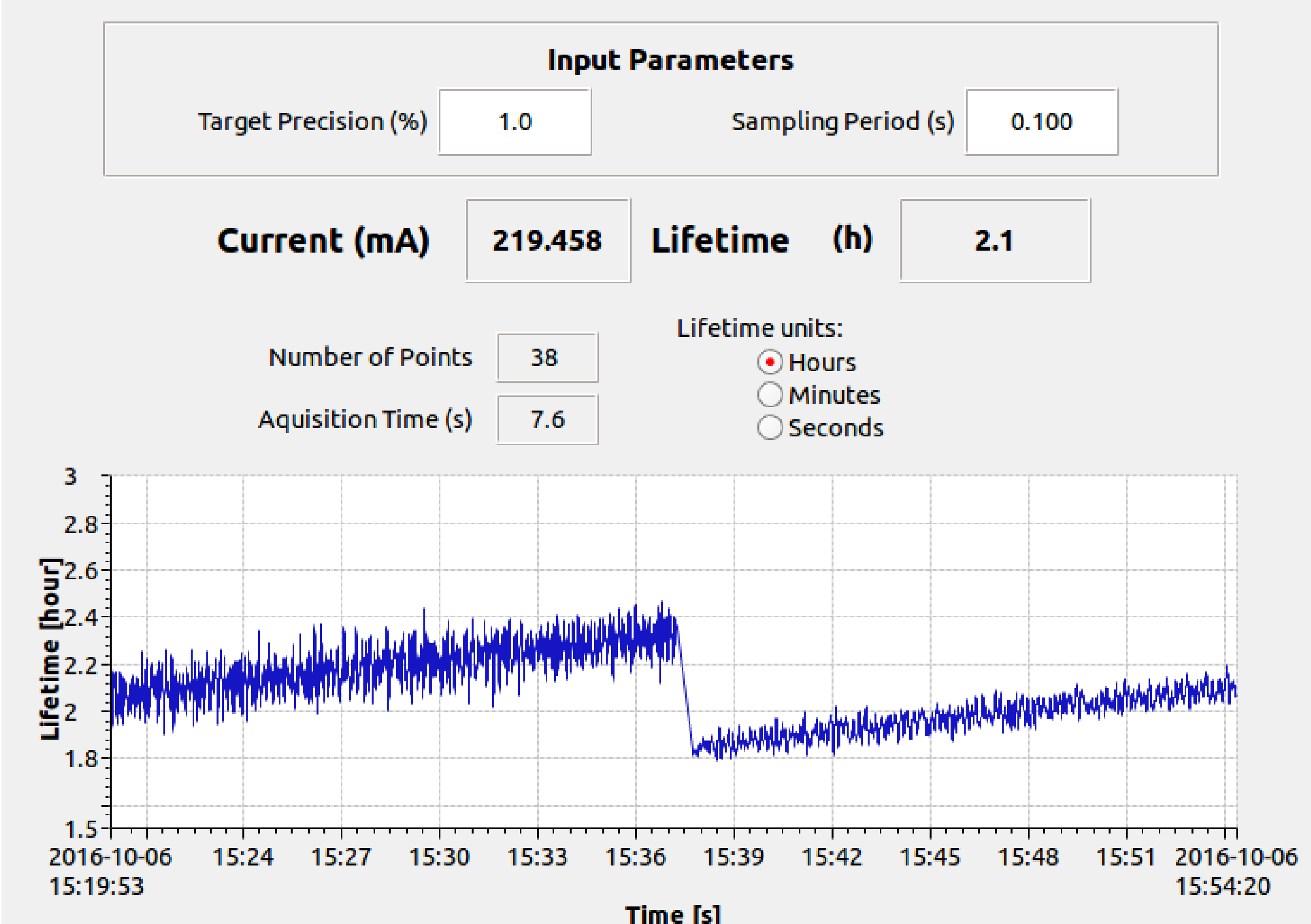


Figure 4: Lifetime display application in the CS-Studio environment.