The BepiColombo Quick-Look Analysis (QLA) system is a web-based application being developed by the Science Ground Segment (SGS) team for the BepiColombo mission. The QLA provides a means to interactively visualise and assess the science and housekeeping data products soon after the telemetry has been received from the spacecraft. This allows a preliminary analysis of the science products produced by the instruments as well as continuous monitoring of the spacecraft and instrument health. By providing quick visibility of the data the system aims to facilitate the monitoring of the completion of the observations with respect to the scientific objectives and to feed this back into the mission planning process.

### Data Visualisation

Data is displayed in the QLA through a number of different dashboard views. Each dashboard can contain multiple widgets showing different types of related data (housekeeping, science) with various plot types.

Each widget in a dashboard is a container with a set of common functionality for visualising and interacting with the data - such as widgets for displaying charts, images, maps or tables.

**Interactive Plots**
- Data is loaded into the web browser which allows real-time interaction with the data
- Data values and metadata are displayed when hovering the mouse over data points
- Plot widget types support zooming and panning.
- Some manipulation of data is possible, such as changing axis type (linear/logarithmic)

**Data Export**
- Data can be exported to CSV files, images or PDFs.

**Data Linking**
- Linking several widgets allows actions performed in one widget to be applied to the linked widgets, facilitating the comparison of data.

**Saveable and shareable**
- the given state of a dashboard, such as time range selected and data processing level, is stored in the URL
- options set on each widget can also be stored in the URL
- this URL can be bookmarked to return to the QLA on a given dashboard in a given state
- this also facilitates sharing data of interest with colleagues!

**Customisable**
- widgets can be added to custom dashboards by dragging them in from a widget menu.
- the layout of the widgets can then be modified as desired.
- this allows a user to more easily compare different data types and across different instruments.

### Web Technologies

- **Angular** - a javascript framework for building client applications in HTML and TypeScript. [https://angular.io/](https://angular.io/)
- **WebGL** - web standard for a low-level 3D graphics API based on OpenGL ES.
- **Font Awesome Free** - Icons available under the CC BY 4.0 license. [https://fontawesome.com/license](https://fontawesome.com/license)
- **Spring Framework** - simplifies and provides a lot of ready made functionality for the QLA server-side code. [https://spring.io/](https://spring.io/)

### Contact:

Alan J Macfarlane
BepiColombo Science Ground Segment (SGS) Software Engineer
alan.macfarlane@esa.int

---

**BepiColombo:** The Mission

**BepiColombo** is a joint ESA/JAXA mission that launched on 20th October 2018.

It will arrive at **Mercury** in late 2025, including several flybys of the Earth, Venus and Mercury along the way.

It consists of two orbiters:
- **Mercury Planetary Orbiter (ESA/MPO)**
  - hosting 11 scientific instruments
- **Mercury Magnetospheric Orbiter (JAXA/MMO or 'Mio')**
  - hosting 5 scientific instruments.

### Performance

Advancements in web technologies mean that visualising and manipulating data in a web browser is a valid possibility. Even so, displaying many thousands to millions of data points is still a challenge.

- to reduce strain on the browser and its memory, data can be downloaded on the server. Higher resolution data is retrieved when a user zooms in.
- to reduce the memory footprint on the server, data is streamed to the client in batches via parallel processing threads.

### Rapid Access to Data

Soon after telemetry has been retrieved from the spacecraft, raw and calibrated products are available to be displayed.

The QLA reads PDS4 files directly, as well data indexed in a database, to be displayed in the user interface.

[QLA](https://www.khronos.org/webgl/)

[plotly.js](https://plot.ly/javascript/)

[Stack Gl](http://stack.gl/)