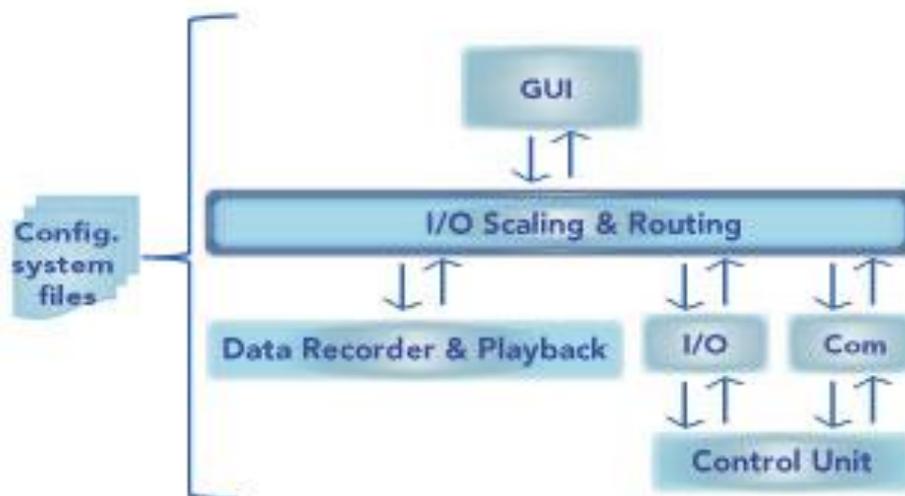


## Viking Software platform

The Viking is Prevas software platform for measurement and control system, designed according to open and modular guidelines.

Viking is highly reconfigurable and not bound to a specific type of I/O, or a specific type of Graphical User Interface (GUI) layout, as this changes from project to project.

When using the platform it's possible to get a system up and running in very short time. The user can directly focus on the project specific parts (GUI, special I/O etc.) since the backbone for the system is already in place with add-ons for standard I/O and communication.



### Config. files

Virtually everything in the system is dynamically configured; Signal names, pins and parameters, add-on configuration, which panels to use etc. All this configuration can be stored to a Config. file.

This allows easy reconfiguration between projects/users etc. The only required Config. files are those that define; signals to be used, the main setup for the GUI and the measurement engine.



## Add-ons

As each customer have different needs; the Viking platform is a compact platform with a number of standard and customised add-ons. An add-on could be ordinary I/O, a specific communication module, or functionality such as HIL. As the software is build with LabVIEW, virtually anything can built in or wrapped in (Matlab, C/C++, VHDL, etc.) using LabVIEW.

Commonly used add-ons:

### HIL

Adds support for models (using National Instruments, SIT 5.X).

The inputs and outputs of the model is automatically loaded, and access to scalar Model Parameters and Model Signals can be defined through aliases. Also supports downloading of large Model Parameter matrixes.

### Test Sequence

Automated testing requires test sequences. Prevas' solution is based on the commercial tool TestStand where test sequences can be made in TestStand, C, .NET ActiveX, or LabVIEW.

## Record and Playback

All types of parameters can be logged, the logging setup defines the data recording rate, as well as the start and stop conditions. Start and stop can be defined with triggering, i.e. started/stopped when a parameter value goes within a specific range.

It is also possible to playback recorded logfile, to perform the exact same test over again, by starting a playback session. The user only needs to specify what parameters to include in the playback. All streaming buffering etc, is handled by the system.

Sequence is independent of the test system and the control unit and each test setup is easily changed as it's defined in the configuration of the system.

## I/O Scaling and Routing

The I/O scaling and routing uses the Config. files to provide a abstract layer for the user in order to use known signal names and parameters during the testing instead of voltages and bytes of data. Scaling handles the translation between the electrical reading (voltage, frequency, bit pattern) and the API functions (speed, temperature, message). Routing links the API's function to a specific physical signal (wire)

This flexibility in both scaling and routing ensures the system to be able to handle changes.

Typical changes are when a sensor model is changed, an I/O needs to be replaced or a new Control Unit (DUT) needs to be tested.

## Dynamic Graphic User Interface (GUI)

The GUI is built with reconfiguration in mind, and unrestricted number of views can be added as plug-in panels. Each panel can be configured to control or present any parameter in the system in realtime. The user can then store the setup for each panel for later use.