

EyesWeb Mobile - Users tutorial

The EyesWeb Mobile framework is an add-on for EyesWeb which let one design user interfaces for EyesWeb patches. It is made of two components called EyesWeb Mobile designer and EyesWeb Mobile runtime.

The common use case is to develop an EyesWeb patch with the EyesWeb Development Environment (EywGui). Then, you import the patch in the designer and draw the user interface. Finally, you run an EyesWeb server on a computer on the network and run the EyesWeb Mobile runtime tool to control the patch (of course, runtime and EyesWeb server can also run on the same computer). Note that the EyesWeb server might be run as a service, thus, the user does not actually need to start it as it is always available as a background process.

The aim of these tools is to design a simple-to-use user interface such that the final user does not need to deal with EyesWeb to run EyesWeb patches. Moreover, in the case of complex patches, the complexity is hidden to the final user as it just sees the controls that have been placed in the graphical user interface.

Step-by-step instructions

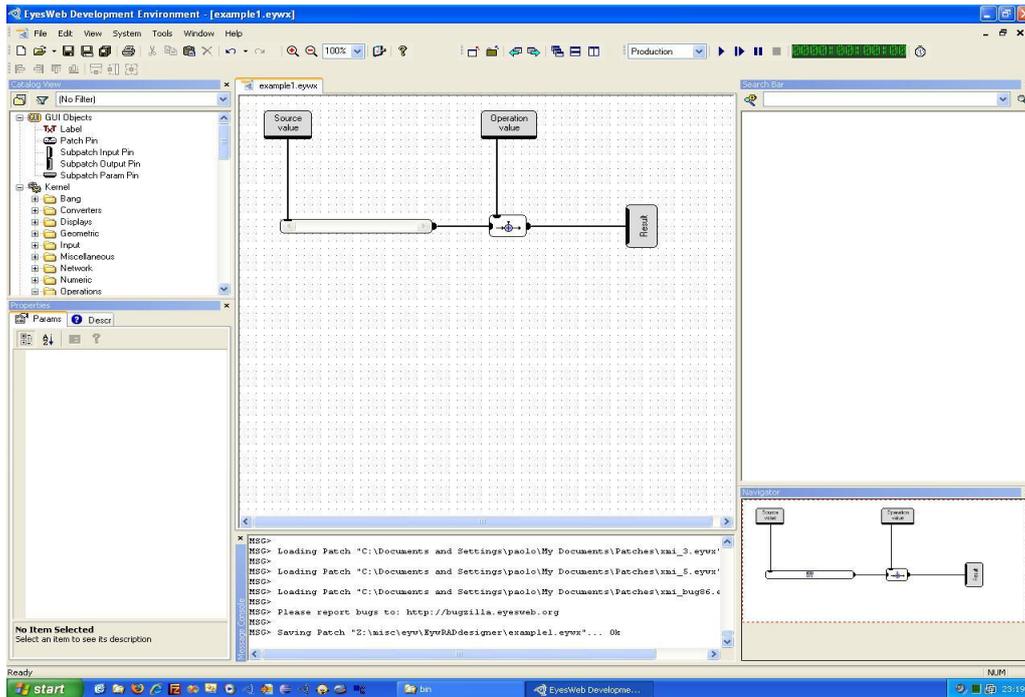
First of all, you have to download the needed software. Since version 5.0.3.0 of EyesWeb, EyesWeb Mobile and some example applications are included as part of the distribution. Thus, you simply need to download and install the latest version of EyesWeb:

ftp://ftp.infomus.org/Evaluate/EyesWeb/XMI/5.0.3.x/*.*

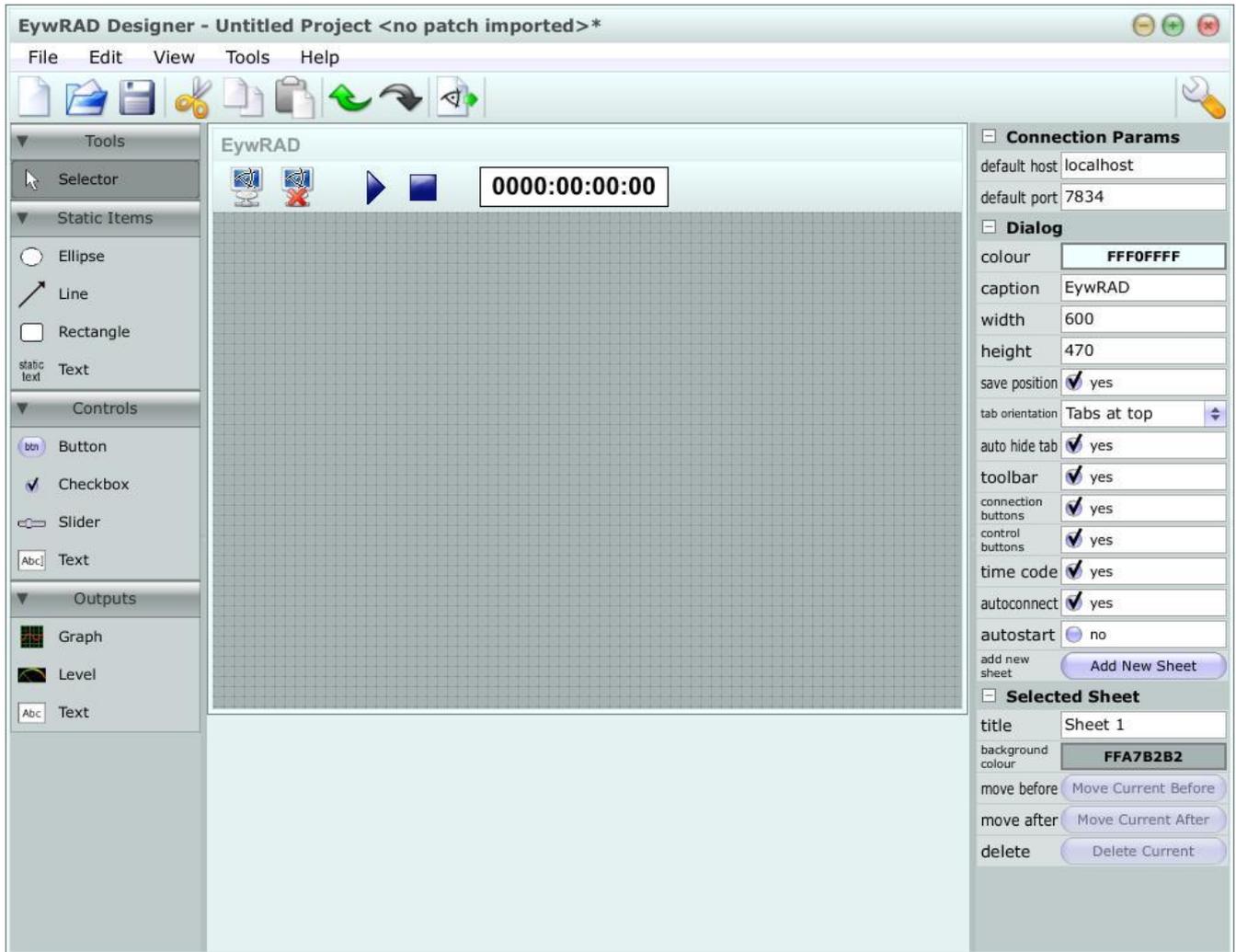
The following section shows how to build a project file (.eywrad) from scratch.

Using the designer

A patch is developed via the EyesWeb Development Environment



Then, the EyesWeb Mobius designer is started and a graphical user interface is drawn for that patch. The following screenshot shows the designer when it is just started, and an empty project is shown.

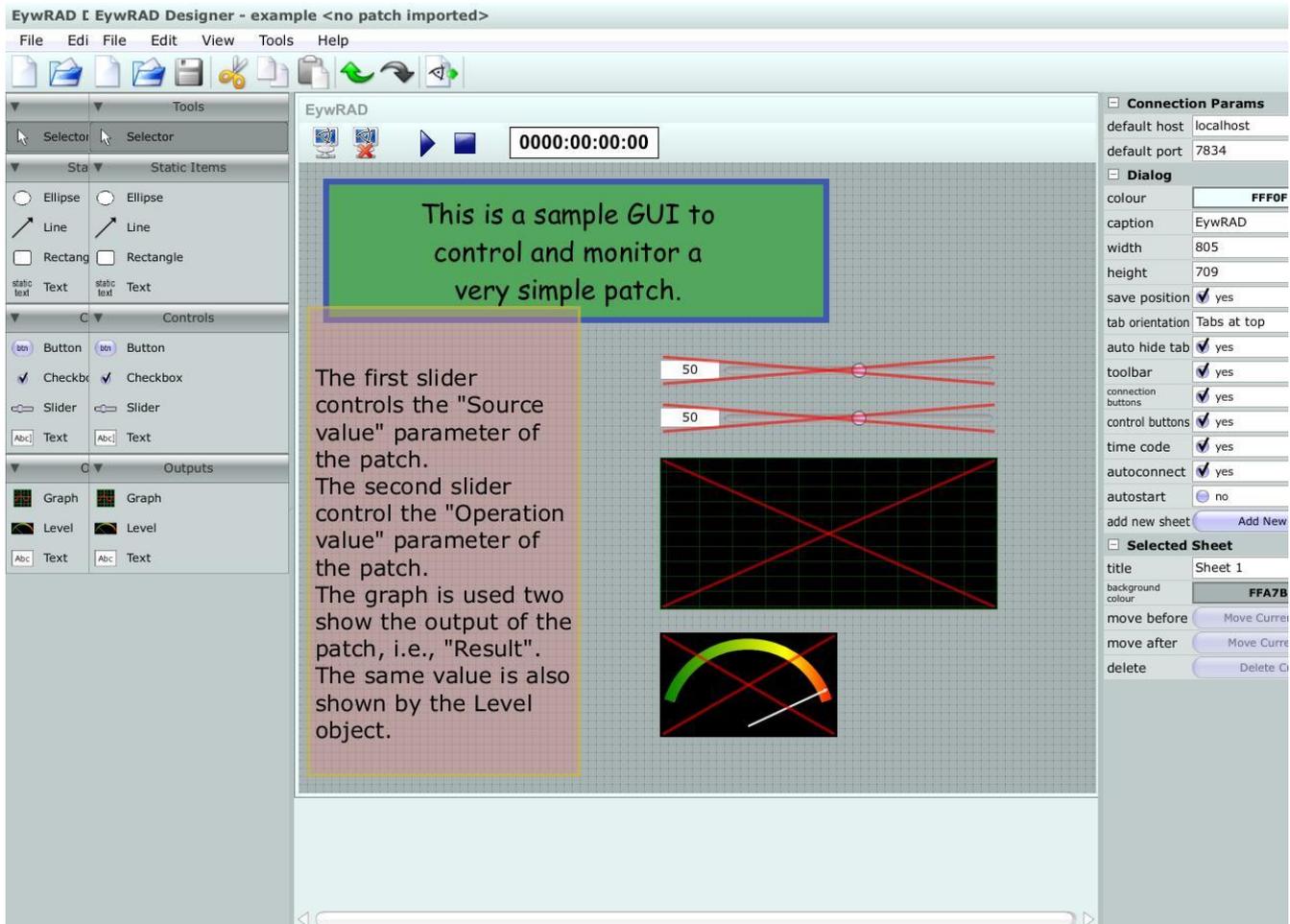


On the left side the available tools are shown. The first group of tools (Static Items) contains objects that are used just to improve the graphical appearance of the drawn interface, or to document it. The second group of items (Controls) represent objects that can be placed in the canvas to control the EyesWeb patch (e.g., a slider may be used to control a numeric parameter of the patch). The third group of objects (Outputs) contains items that are used to monitor the outputs of the patch (e.g., a graph can be used to visualize a numeric result of the patch).

On the right the properties of the selected object are shown, thus, the available items depend on the selected object. In the screenshot, since no item has been selected yet, the properties of the application are shown. Finally, the central editor contains the application that is being drawn, i.e., the application that the final user will use.

In the example patch we are using (which is very simple) we might control the two parameters connected to the subpatch parameter pins (Source value and Operation value), and we might monitor output of the scalar operation. Thus, we design an interface which contains two slider, and two outputs. The slider control the two parameters, respectively. The two outputs (a graph and a level) are used two show the result of the patch: both objects show the same value, but with different layouts.

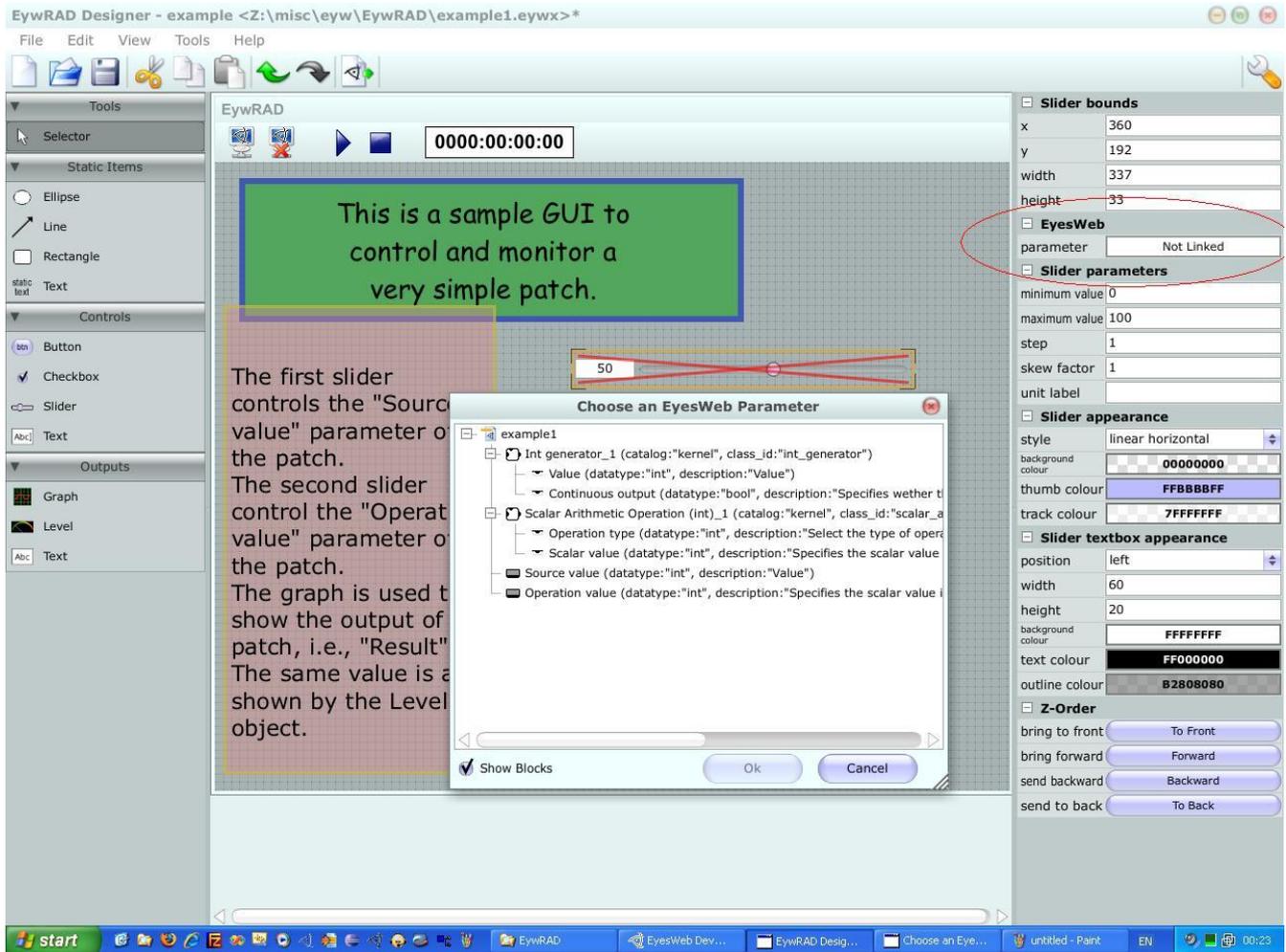
In the following picture the needed items have been placed in the canvas; besides them, some text items have been placed in the editor to explain the application to the final user.



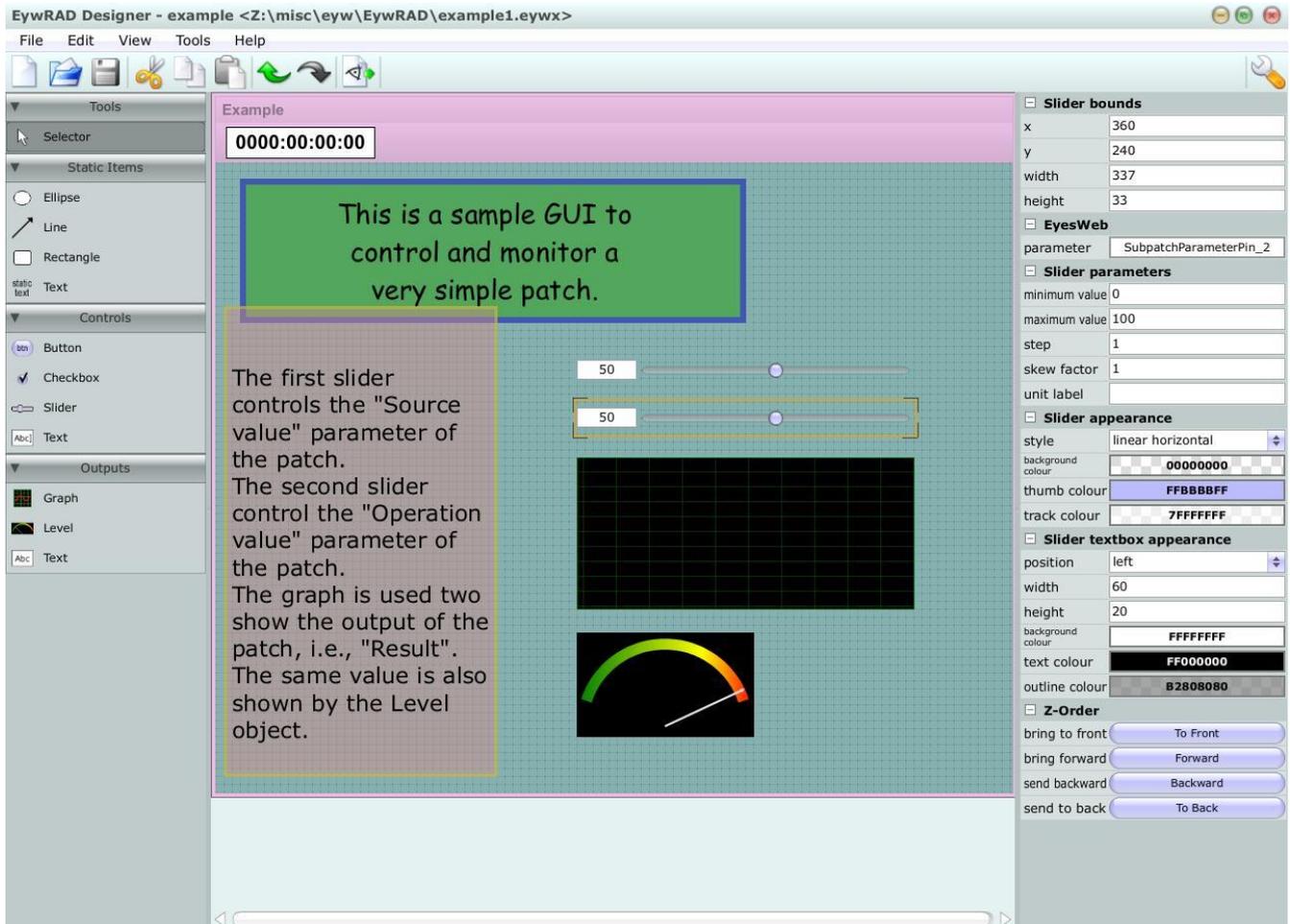
Note that the control and the outputs are shown with a red cross over them; the reason is that they have not been bounded to the underlying values of the EyesWeb patch. As a matter of fact, up to now the designer has no info about the patch we plan to work with, thus, we now need to import such patch. the import operation is needed to let the designer know the parameter and outputs which are available.

To import the patch we simply click on the  button and select the patch. Now we can perform the bindings.

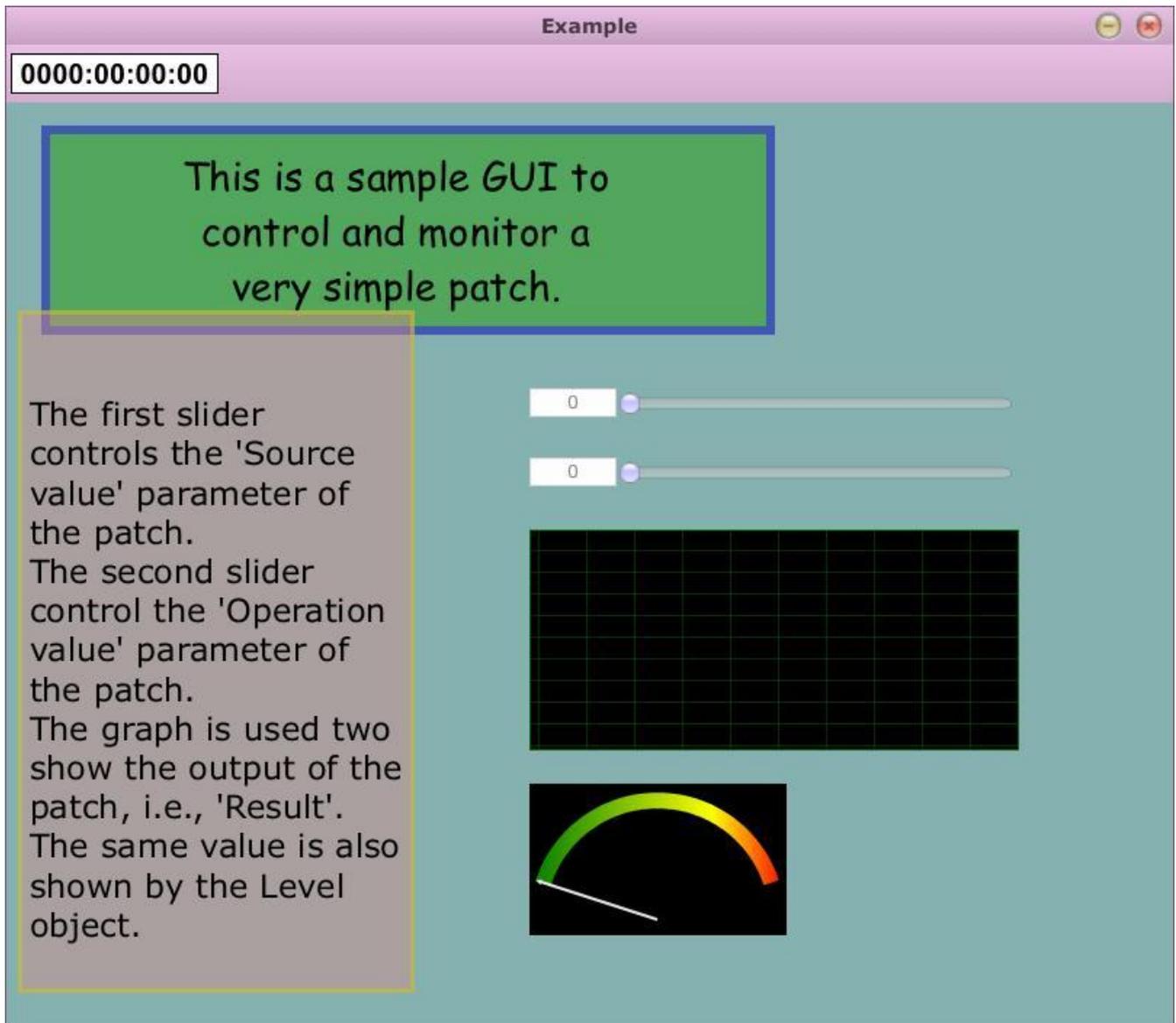
We select the object we want to bind, e.g., the first slider; in the right pane, among the available properties we click on the *parameter*, in the *EyesWeb* section (circled in red in the following picture). The *Choose an EyesWeb parameter* dialog will appear. In this dialog we can choose the EyesWeb parameter we want to bind to this control. In this case we choose *Source value*.



We perform the same step for the other controls and output, and the red crosses will disappear. Finally, we perform some finishing touches, such as to change the colors, to set the caption of the application and so on. Note that we also simplified the toolbar available to the user: we removed the connection buttons and the start/stop buttons. The user will only see the timer (of course, this might be removed too if one wishes): connection to the EyesWeb server and patch start/stop will happen automatically. The result is shown in the following picture.



Finally, the only remaining step is to save the project and to run it with the EyesWeb Mobius runtime. The runtime will automatically connect to the server and will run the patch: the following picture shows the runtime at work.



Installing and running EyesWeb Mobius under Windows CE

Copy the eyesweb_mobius.cab file to your mobile device (e.g., using ActiveSync). Run the cab file (this will install the software on the mobile device); you should now have a link to EyesWeb Mobius in the Start menu of your mobile device. Run EyesWeb Mobius from such link; it will ask a name of the eywrad project to be used, select it (copy the projects designed with the EyesWeb Mobius designer to your mobile device by using Active Sync) and EyesWeb Mobius will run as usual.

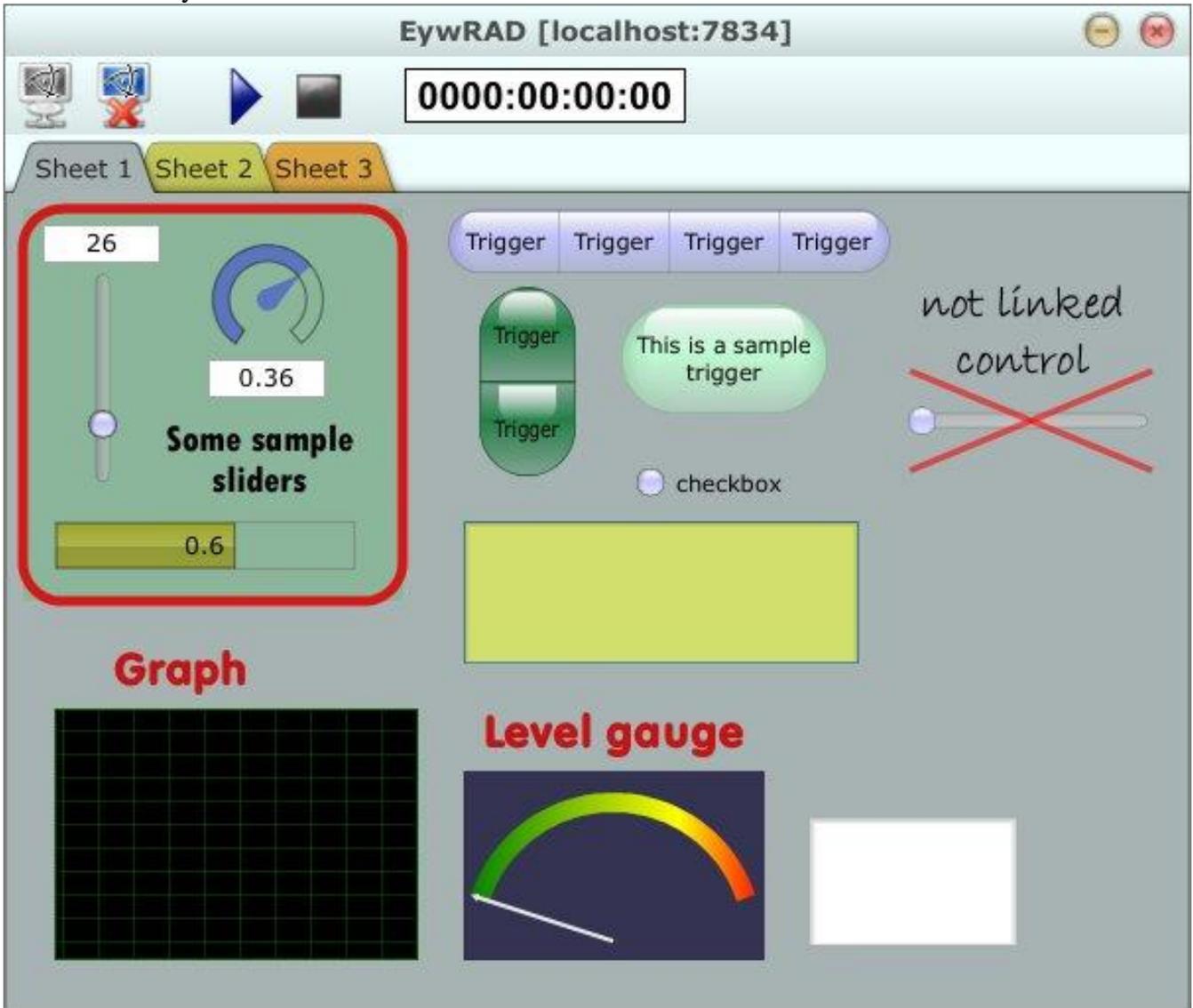
Notes on Windows CE

Many devices (e.g., some Dell Pocket PCs) disable the wireless connection when placed in the cradle. This might the connection to EyesWeb server (EywConsoleRCF) to be unavailable. Remember that the computational power of such devices is much limited if compared to standard

PCs (desktops or notebooks); take all necessary steps to reduce the processing requirement on your mobile device. Moreover, the network latency is higher than standard computer connected via WiFi. As an example, if you want to display a video stream, check the size of the video tool in the designed GUI, and reduce the image to that size in the EyesWeb patch (which runs on a more powerful PC). This avoids the resize operation in the mobile device, and also reduce the used bandwidth as a smaller image is sent via the WiFi connection.

Screenshots

The following screenshots are unrelated from the above example. They just provide some more views of the system at work:



Choose an EyesWeb Parameter

- ttt
 - Int generator_1 (catalog:"kernel", class_id:"int_generator")
 - Value (datatype:"int", description:"Value")
 - Continuous output (datatype:"bool", description:"Specifies wether the output is always generated (true) or only")
 - Input selector_1 (catalog:"kernel", class_id:"input_selector")
 - Sample Rate Limiter_1 (catalog:"kernel", class_id:"sample_rate_limiter")
 - Clock mode (datatype:"int", description:"Specifies which type of clock is used to compute the time elapsed betw")
 - Screen capture_1 (catalog:"system", class_id:"screen_capture")
 - Frames per second (datatype:"double", description:"Frames per second")
 - Display_1 (catalog:"system", class_id:"image_display")
 - test_param_name (datatype:"int", description:"test_param_desciption")

Show Blocks

Ok Cancel

<http://www.infomus.org>

<http://www.eyesweb.org>

<http://www.trilcenter.org>