

MGL iEFIS Rotorcraft setup



This guide applies to MGL iEFIS systems from firmware version 1.0.1.9 or higher.

General

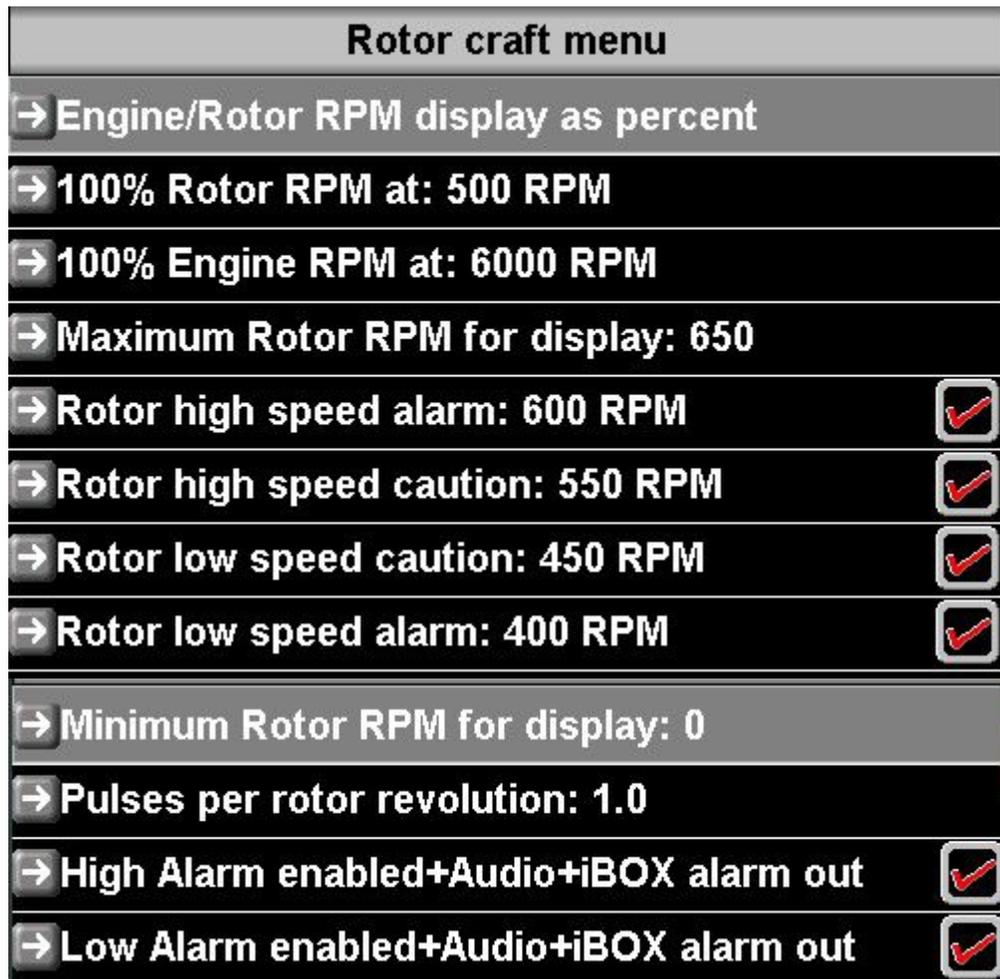
The iEFIS provides two rotor speed inputs on the iBOX marked R1 and R2.

These inputs are designed to be interfaced to a sensor that provides a square wave signal in the range of 0 to 6V (switching between 0V and 6V). The “high” voltage may be higher – up to 24V but should not be much lower. Lower level should be less than 0.5V.

The iEFIS Lite does not have an iBOX. For the “Lite” you can obtain rotor speed from the RDAC XF RPM 2 input, leaving RPM1 for the engine RPM.

If you would like to use the Lite in this way please ensure that the pulses/rev setting for engine RPM 2 is set to 0.0. If this is the case the EFIS will use the RPM 2 input as rotor speed sensor input and you set the pulses per rotor revolution in the Rotor craft setup menu. If the RPM 2 setting is not 0.0 then it will remain an input for engine RPM and the Rotor RPM indication will remain at zero.

The Rotor craft setup menu



All rotor craft related setup functions are combined in this menu.

Most items relate to visual displays like bargraphs.

Rotor and engine RPM may be displayed as actual RPM or as a percentage relative to a set limit that you define here (the 100% RPM value).

Creating a visual display for a rotor craft

The EFIS does not have any form of visual rotor craft related display as it is delivered from the factory. Rotor craft avionics installers may provide you with a set of modified screens that do contain such displays or you can create your own.

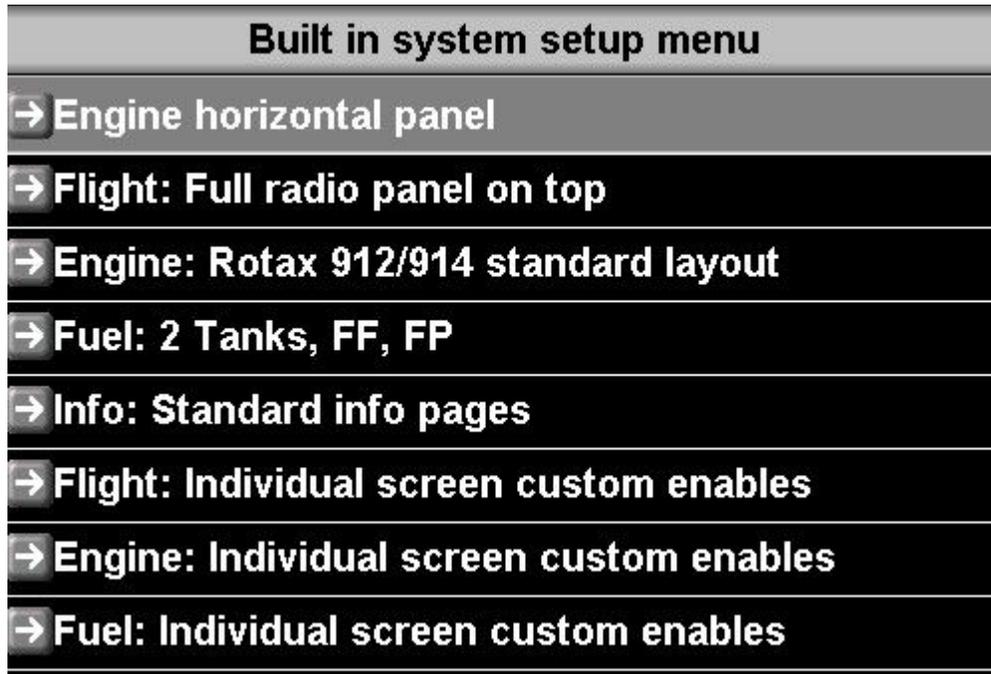
To create your own you use the iEFIS screen designer and simulator which is a free download from the MGL Avionics website.

Consult the document "iEFIS alteration guide" which is also a download from the website. This gives details on how to use the screen designer.

This document gives a step by step example. You may of course use this or use it as a guide for your own creation.

Step one:

Install and run the screen designer. In the system setup menu, find the item “Standard system selections”. Make sure a flight is not active and airspeed reads “0” in the simulator or else it may refuse to open this menu.

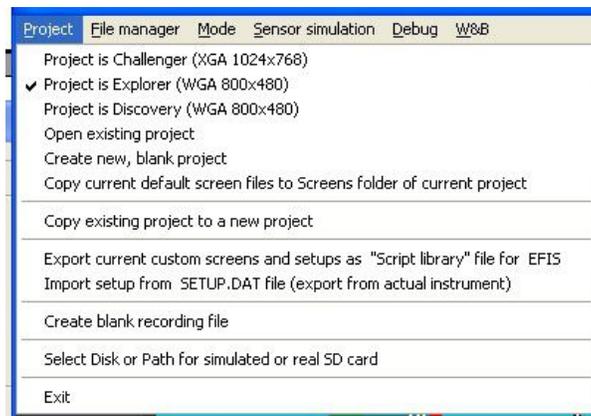


Here we will use this built in screen choice on an Explorer EFIS.

Ensure that all “Individual screen custom enables” are set to “set by selection”.

OK so far ?

Now exit these menus to get back to the normal screen, in this case screen 1. Open the project menu.

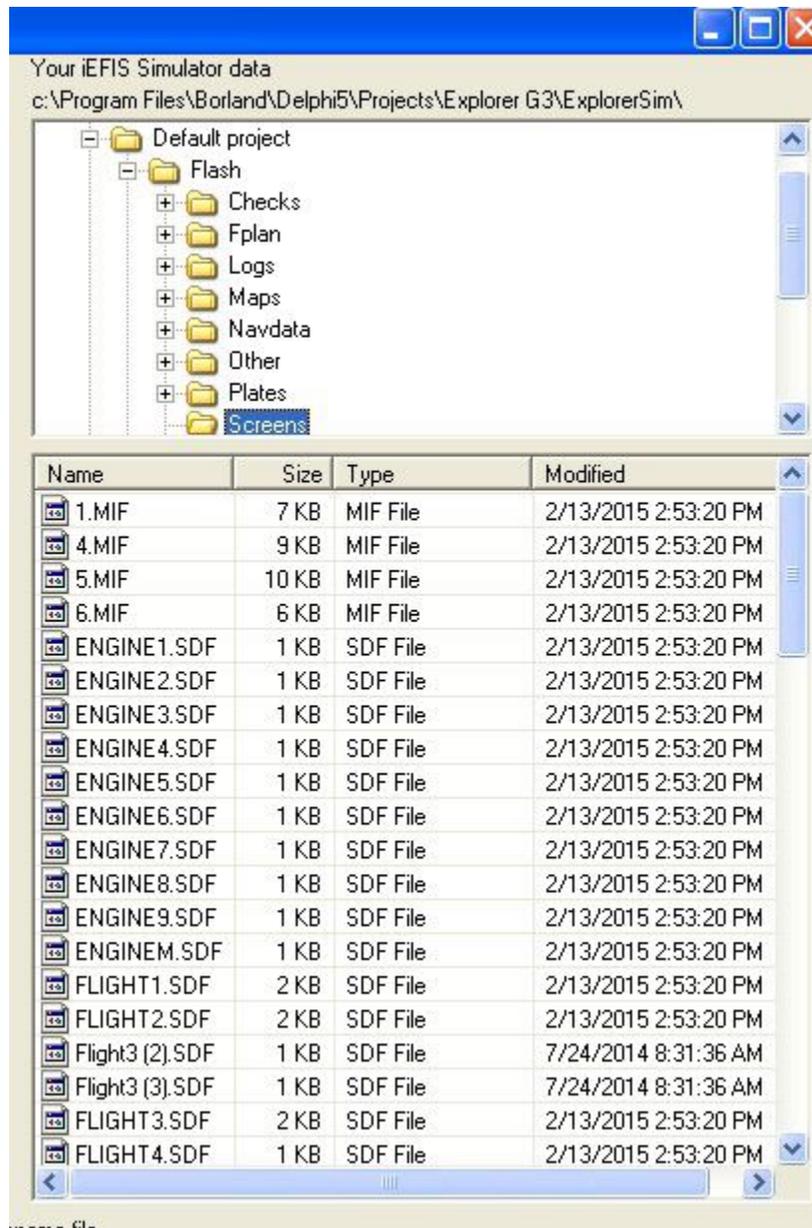


Choose “Copy current default screen files to Screens folder of current project”. This will now take the built in screens you have chosen to run in the previous step and create a bunch of screen files in your current projects “Screens” folder. You can create a new project for that or

just use the “Default project”.

That is just a name for a folder on your system where a simulation of an actual EFIS internal disk is located – you will work with this later.

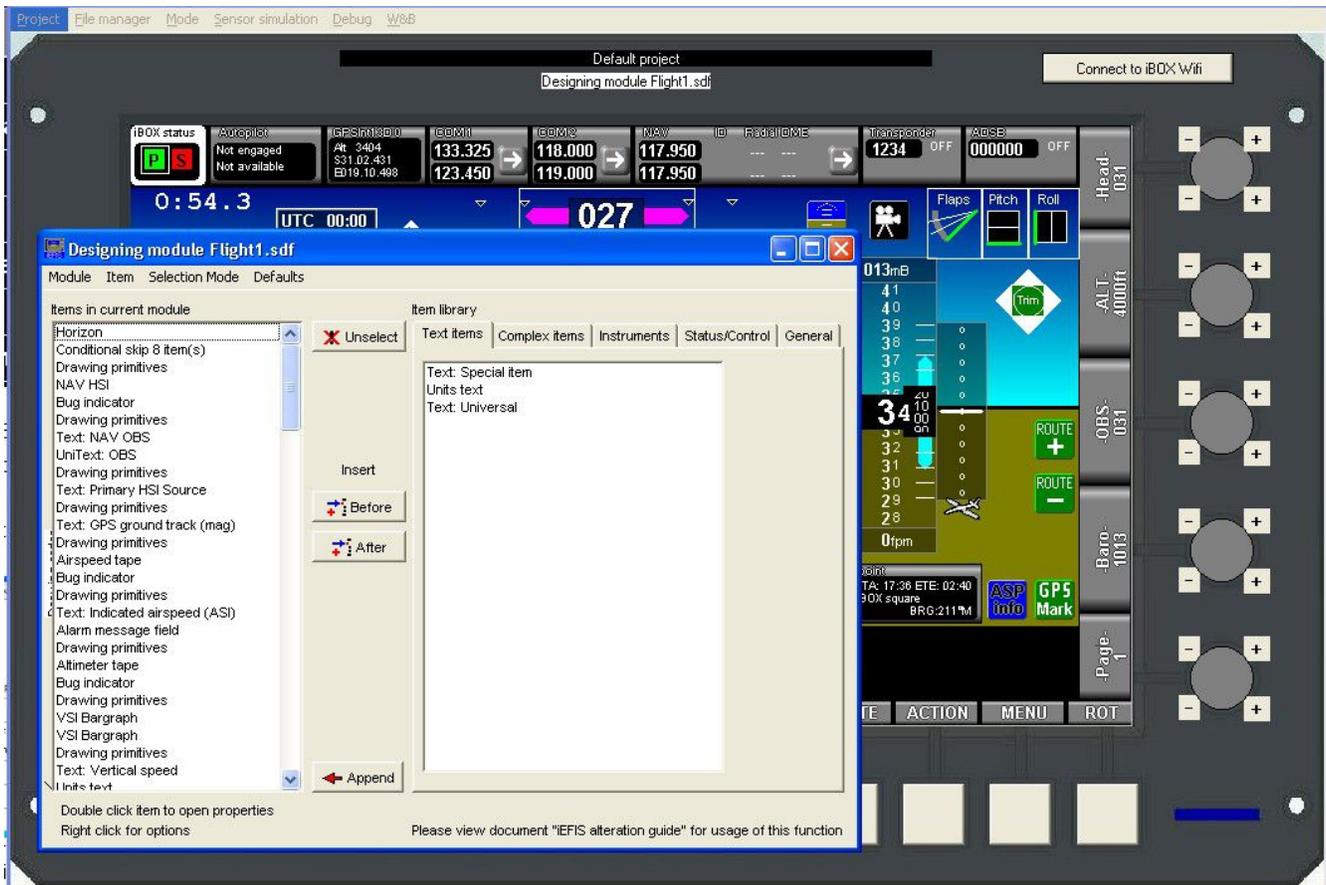
You can use the file manager to view the Screens folder of the current project – it should look like this:



Scroll down the list of files – we are going to edit Flight1.SDF. This file contains a description of what screen item (like an altimeter for example) is where on your screen.

Don't worry – you will not have to attack the file directly, we will use the screen designer to make this easy.

Leaving the display showing screen 1, choose “Mode” and then “Design”.



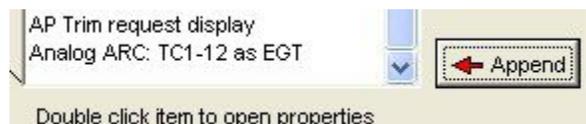
From “Module” choose “Design custom flight module”.

This opens the Flight1.SDF file in your screens folder for editing (or if it does not find that file it creates a blank file that you can start adding items to).

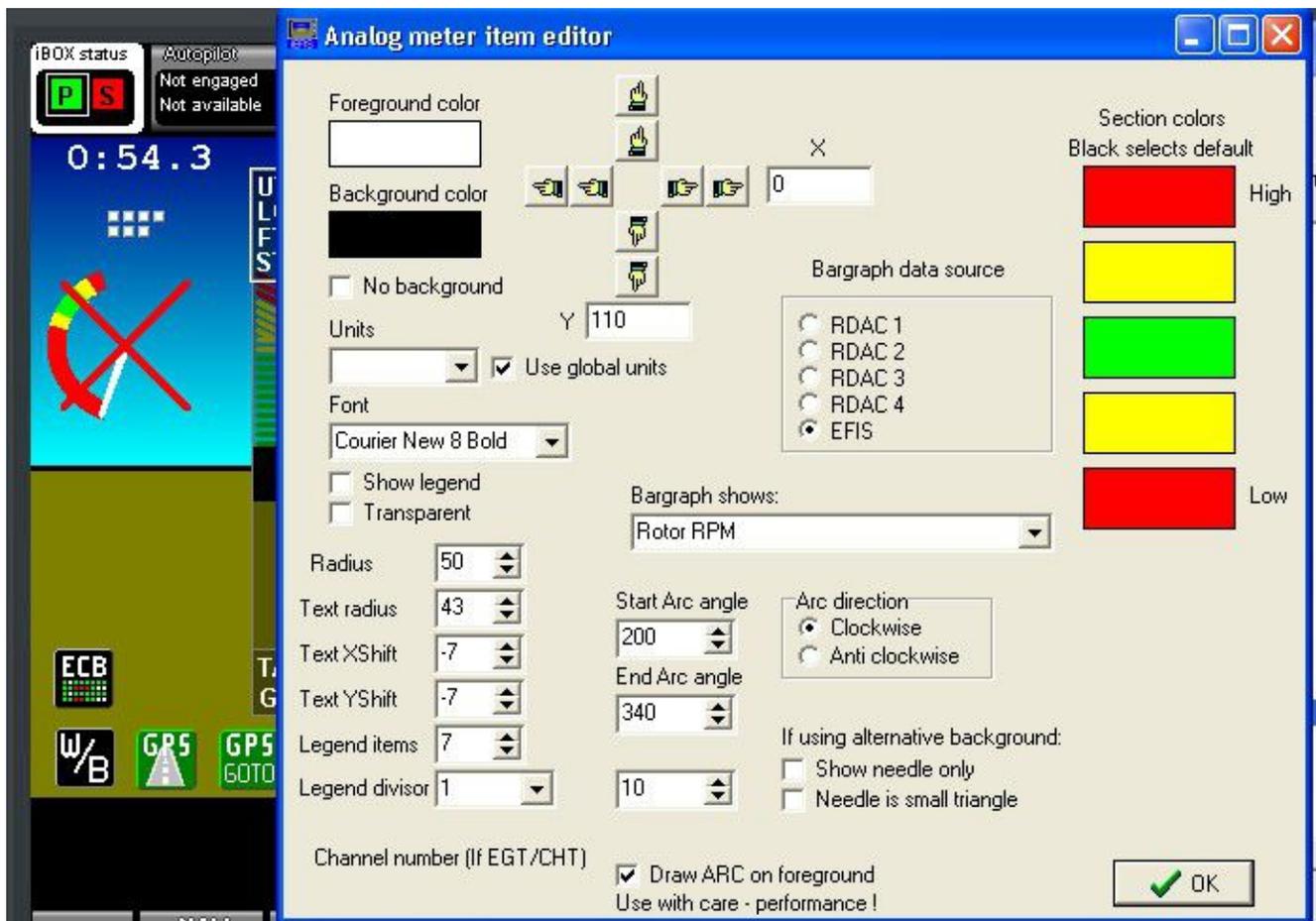
The field on the left shows all of the items in that file in the order that they will draw (this is important as one item could cover another if drawn in the incorrect order).

We are going to design a split needle instrument showing both rotor as well as engine RPM and we will use analog arcs for the display. We will then also add numeric readouts that can show either RPM or percentage as the use of the instrument can choose.

From the “Instruments” tab on the right choose “Analog meter (ARC)” and then click on “Append”. This adds the item to the end of the list.



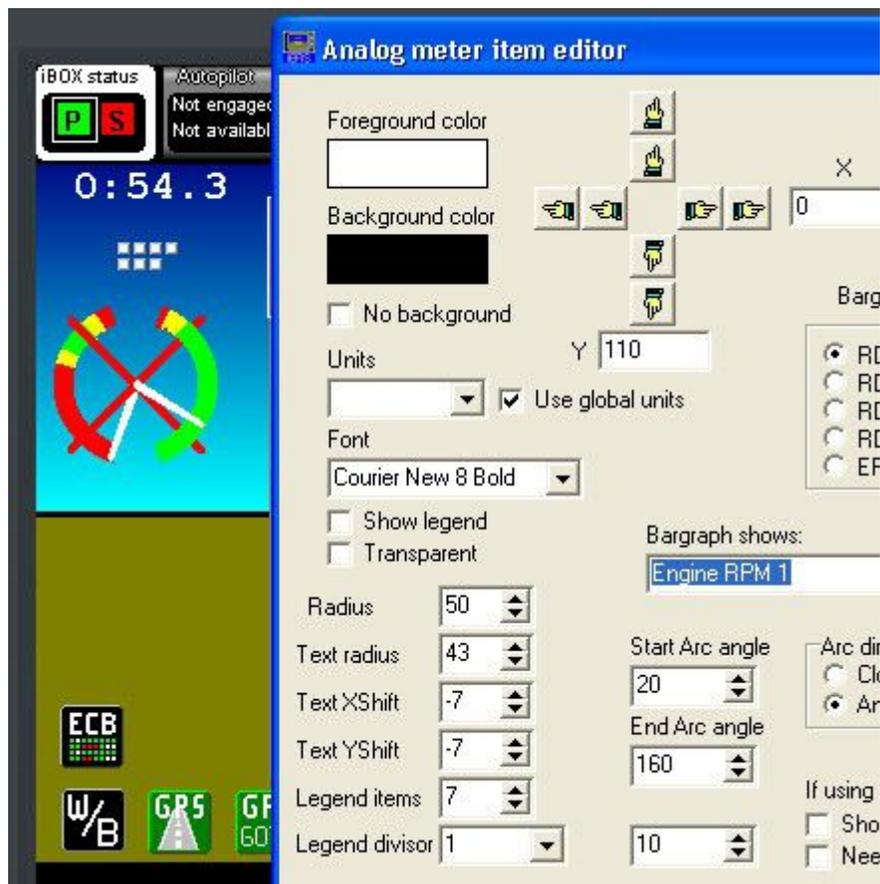
Now double click on this “Analog ARC” text to open the property editor for this item.



Set the property editor like you see above – data source is “EFIS”, item is “Rotor RPM” and draw ARC in foreground. Use the hand buttons to move the ARC so it appears to the left of the airspeed tape, set start and end angles. Do not worry about where the colors start and end – that is chosen in the rotor craft setup menu.

When done, close the property editor.

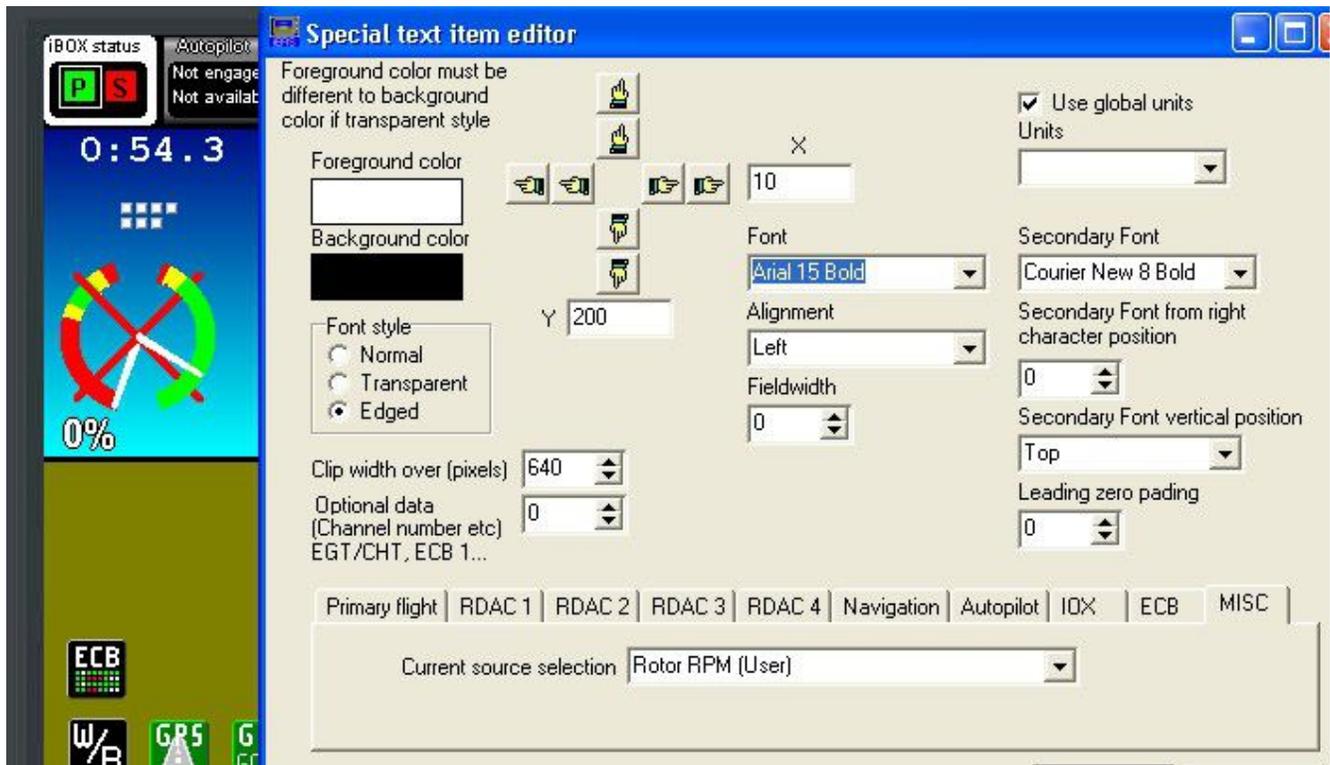
Now add another one and attach that to RDAC 1, Engine RPM 1 and move so we get something like:



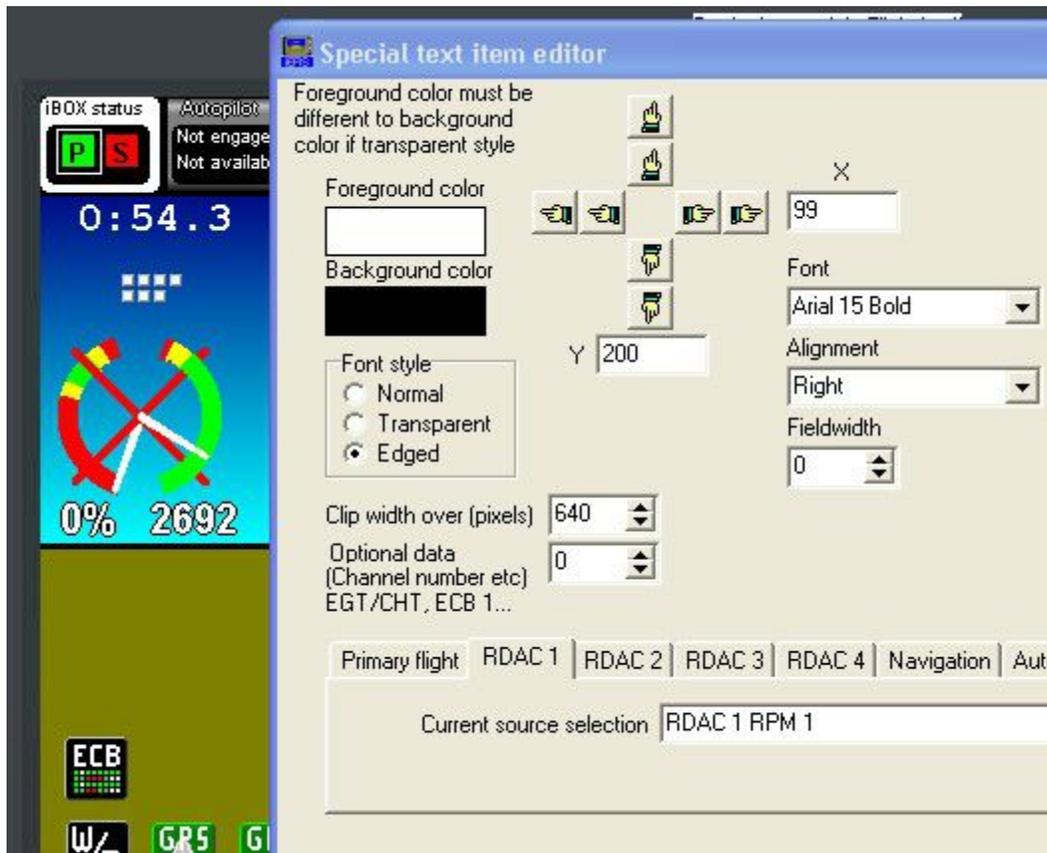
Note that we have selected that this needle goes anti-clockwise.

So now we have Rotor RPM on the left and Engine RPM on the right.

Now let us complete this with two “special text” items that will give us a numeric readout.



Here we placed a use definable rotor RPM field. We selected to show in percent in the Rotor craft setup menu so that is what we get. We could also choose fixed readouts either in RPM or percent as this is listed under "Current source selection" or we could even place both. Note we selected a nice font and choose "edged" font with a black lining to give nice contrast.



Here we have added the engine RPM. Note that we used “right alignment”. This means the text for the engine RPM grows in size from right to left.

Close the designer and we are done. No need to save anything.

You will notice, as you close the designer that screen 1 still does not show your creation. This is because it is still set to show the built in default screens.

No problem – go to the “Standard system selections menu” and now you have two choices:

You can set “Flight: Custom screen designs” or you can set “Flight: Individual screen custom enables”.



The first choice means that it will use the Flight part of any screen from the screens folder ignoring ALL built in flight screens, the second choice gives you finer control on a per screen bases. You will probably want to use the first option in this case as you probably want the new display on all screens.

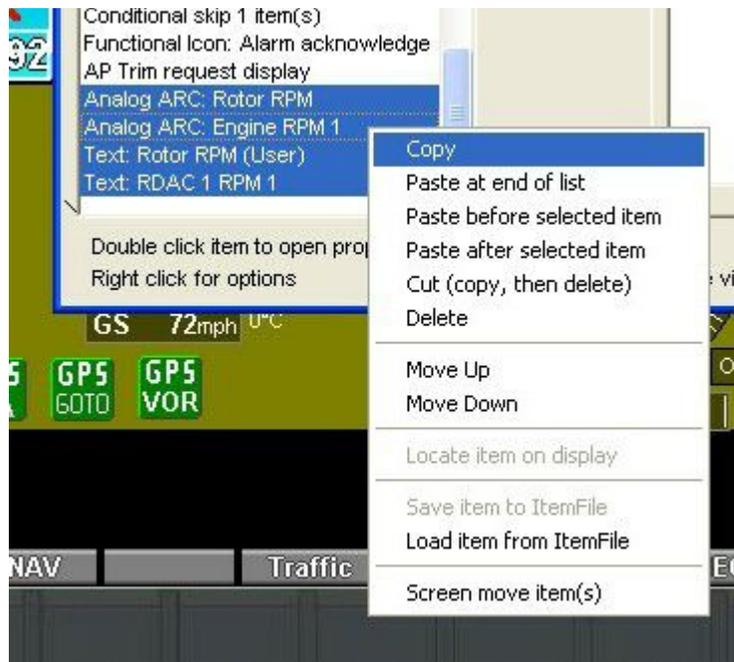
Flight: Individual screen custom enables

- **Screen 1: Custom design in screens folder**
- **Screen 2: Set by selection**
- **Screen 3: Set by selection**

Once you have allowed the EFIS to use your custom screens they will show as expected. Please note that on the real EFIS you must do exactly the same after you copied your new screen files into the screens folder of the real EFIS.



Now, the only thing left to do is to replicate this on all screens. There is a quick and easy way. Open the screen designer again. Edit your Flight file as before. Highlight the last 4 items on the left (click on the 4th last and then click on the last while holding the Ctrl key down to select more than one item).

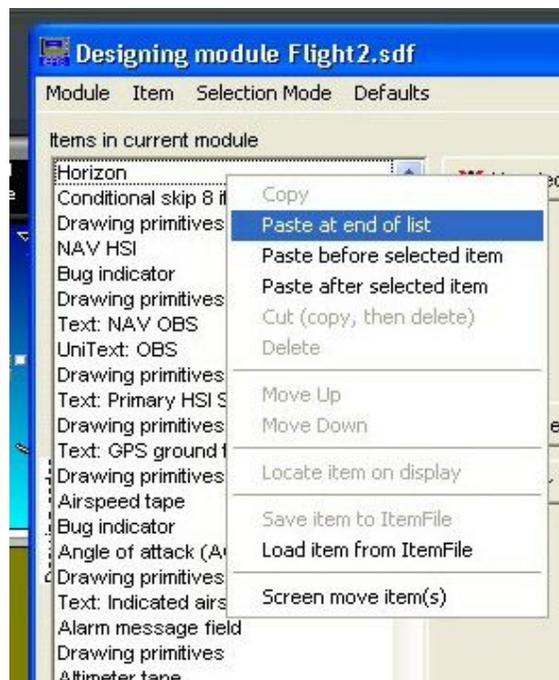


Now right click and from the menu that appears choose “copy”.

It will tell you that 4 items were copied to the clipboard. While you are here, notice other functions which are useful. You can move highlighted items up and down in the list (drawing order), you can copy and paste items and there is a “screen move item(s)”. This brings up a small display with the familiar hand buttons. This allows you to move many items at once on the screen. You will likely use this a lot.

Now close the editor and change to page 2 of the EFIS.

Then open the editor again and choose to edit a custom flight module.



Notice we are now editing Flight2.SDF. Right click and choose “Paste at end of list”. This will

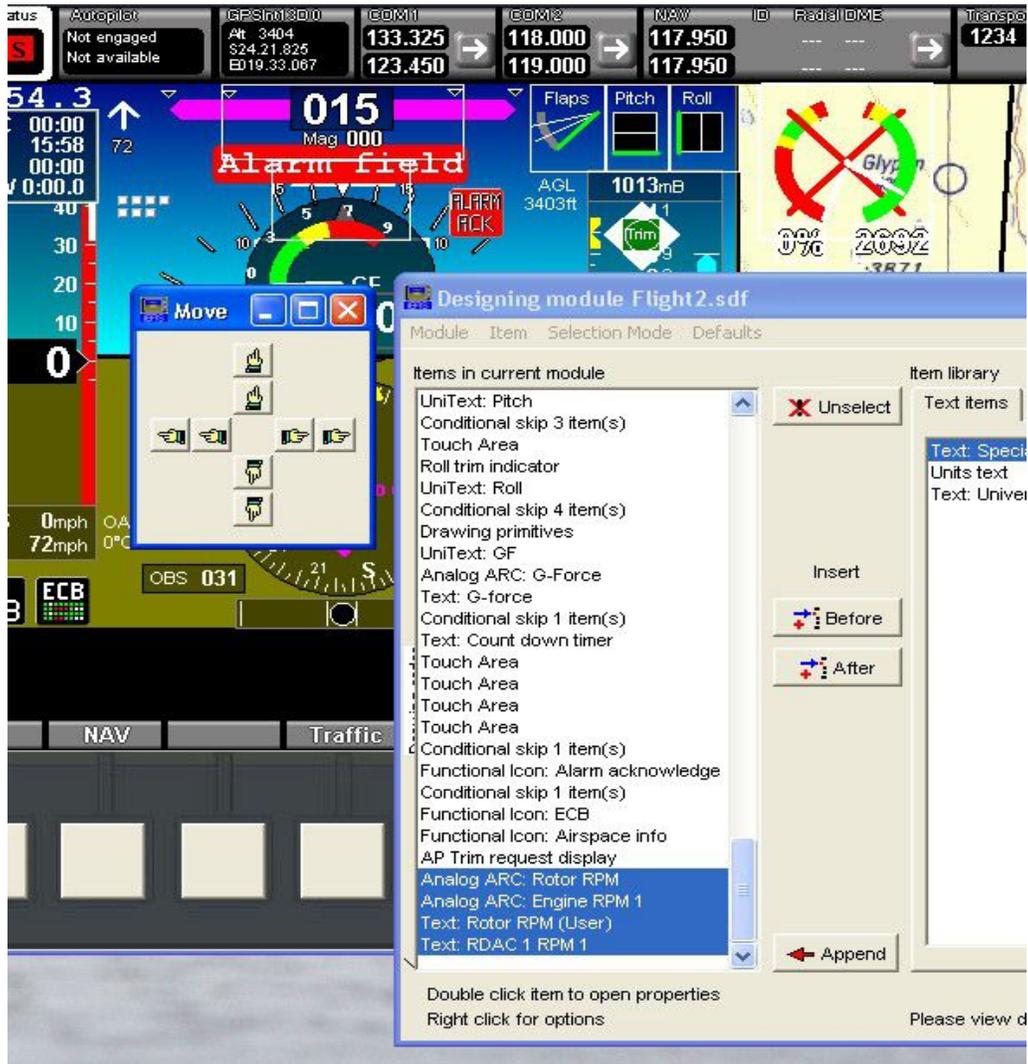
copy the 4 items to the end of the list – but at the same position.



Page two has a map on the right. Let us move our new creation there.

Highlight the 4 items and then right click and choose “Screen move items”.

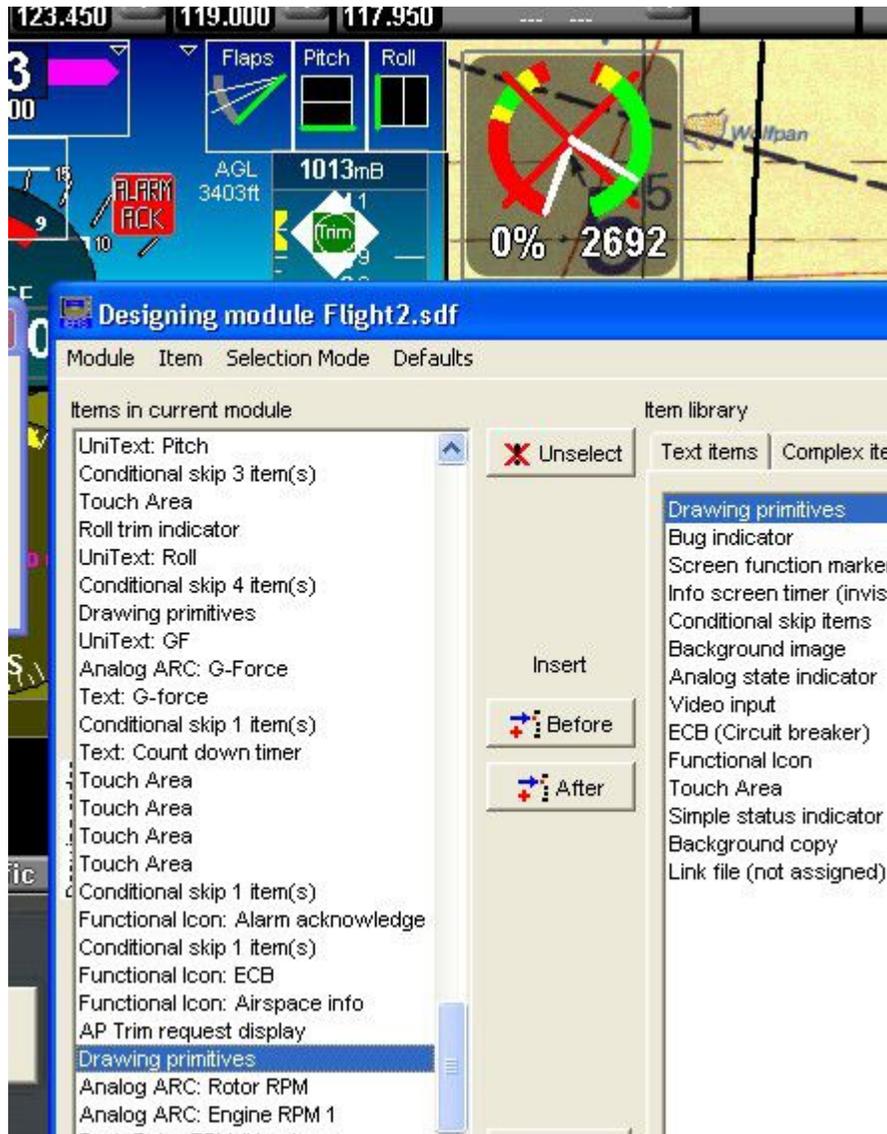
Place the designer and screen move items box so you can see the EFIS screen and move the new items to the map as shown:



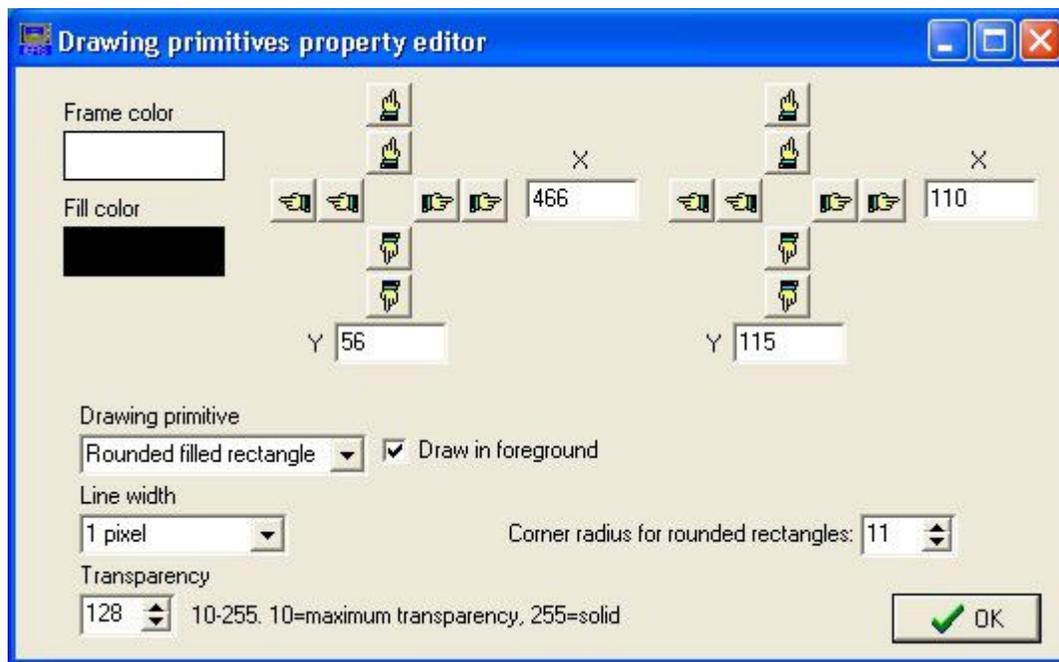
You can repeat this for all other screens.

You will notice in our attempt to move the items over the map that it does not look good as the map has quite bright colors that interfere with the needles.

Let us fix that by placing a semi-transparent rectangle with rounded corners under the new items. The rectangle is a “drawing primitive” in MGL speak. This is just another component you can select from the “General” tab in the designer.



Note that we placed the drawing primitive before the 4 new items in the list. We need to draw that first otherwise it will cover our new items.



Here is an image of the property editor for the drawing primitive.

Note that we set it to draw on the foreground and we gave it about 50% transparency so the background (in this case the map) can show through.

We need to draw on the foreground as the map is redrawn all the time.

Of course, we need to be aware that everything we add to the list takes time to actually draw as it has to be drawn at a high frame rate to ensure smooth changing images.

Transparent items are “expensive” in drawing terms as each pixel needs to be read, then blended with a new pixel and then written back. Normal drawing is done with writes only and this is very fast as the processor can do other things while some other part of the system takes care of moving the data to memory.

However for blending that does not work – the processor must wait for the pixel to be read to arrive from memory before it can do something with it.

Use transparencies sparingly.

The real EFIS provides a function that tells you how much time it takes to draw a screen. It tells you this as a percentage of available time to ensure smooth image creation. You can enable this display in the operations setup menu (it stays enabled for the current power cycle). Try and aim for a value never more than 80% if you have an artificial horizon with synthetic terrain vision and all options on showing. You will likely see a number much less.

This final check should form part of any screen design to make sure you have been using resources efficiently.

Getting the new screens into the real EFIS

Once you are happy that your screens look and operate the way you want them, you need to copy them to a real EFIS.

There are several ways to do this:

You can create a “script library” which gives you many options of what to include. This creates a single file that you can copy onto an SD card and then you use the EFIS function “Execute script library file” in the “Install tasks” menu.

Most of the time, you may just want to copy the screen files in your projects screen folder to a SD card and then in the EFIS you can use either the Install tasks menu function “Copy screen files to screen folder” or you can use the EFIS file manager to copy the files yourself.

The built in file manager of the simulator/screen designer can be used as a handy tool to copy your files to an SD card. You may need to purchase a USB based SD card reader (very inexpensive) from a camera shop or computer shop if your PC does not have a built in SD card reader.

In either case, after you have copied your screens files into the real EFIS, do not forget to allow your EFIS to use the new screen files using the options in the “Standard system selections” menu.