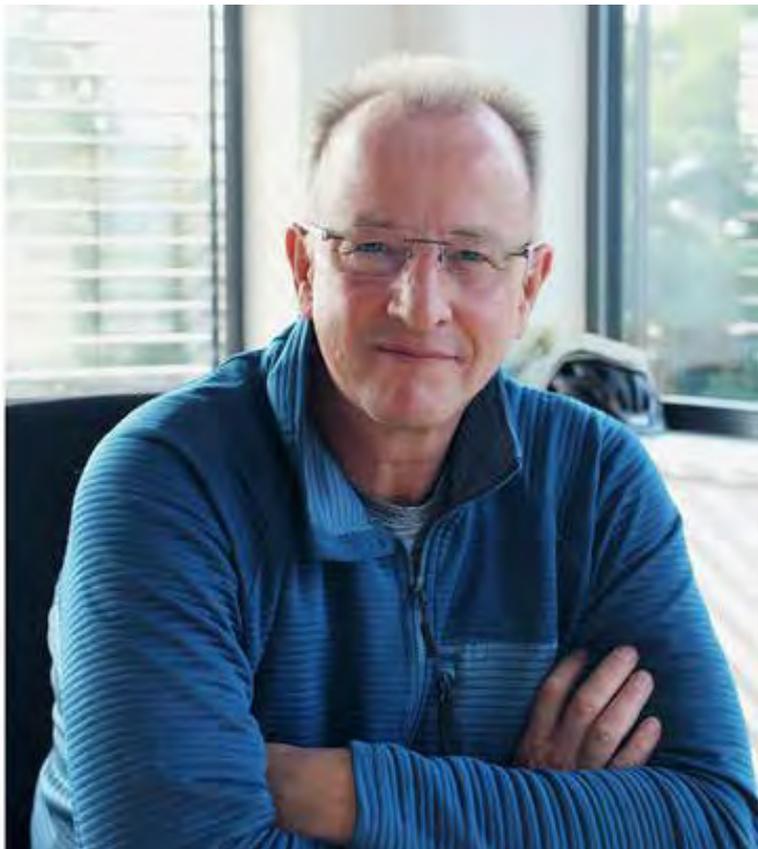


With »emWin« library

User interaction like on a smartphone

3. Februar 2022, 13:30 Uhr | [Tobias Schlichtmeier](#)



Joerg Ehrle is product manager at Segger.

Increasingly, touch displays with graphic elements are found on embedded devices. To program them, comprehensive libraries are needed. One of them is emWin from Segger, which was recently licensed for microcontrollers from Renesas.

Jörg Ehrle is product manager at [Segger](#) and in his function responsible for the embedded graphics library »emWin«.

Markt&Technik: Mr. Ehrle, why are GPUs on microcontrollers (MCUs) and the associated software increasingly important in the embedded industry?

Joerg Ehrle: After the triumphant advance of smartphones, user guidance by means of buttons, rotary switches and the like comes across as somewhat old-fashioned. Think, for example, of cars, washing machines or industrial control systems: There you will find more and more touch displays on which customers then expect user interfaces like on a smartphone.

On a Cortex-M controller, for example, you don't have the resources of a smartphone, neither of processor power nor of memory. With emWin, you can implement interactive interfaces even on resource-limited MCU-based systems. Moreover, emWin can use any hardware and any display.

We have performed different performance tests and talk about megapixels per second on a standard MCU with Arm Cortex-M4 CPUs. You can read all the results and also the memory requirements in ROM and RAM in detail on our website.

Programming graphics libraries is not trivial and the more complex and diverse the elements are, the more challenging it is. What do you say to the developer community about this?

Basically you are right. To program emWin you can take any ANSI C/C++ development environment, so you don't have to leave your usual IDE. However, with our AppWizard you don't even have to program anymore, but you can design the interface visually. This is even more important, because in many companies the GUIs are not designed by programmers, but by graphic designers, who usually do not know how to program. The video »AppWizard Features« on our website shows how easy it is to do this.

As far as we know, there is no tool with a comparable variety of possibilities to scale images & fonts according to needs as it is possible in AppWizard. By default, it converts images for the hardware in such a way that it achieves the best possible performance. But performance doesn't always count, resource hunger does as well – that's why we offer maximum flexibility.

Images and fonts are standard elements of every GUI, how do you differentiate yourself from competitors here?

With the image format, we offer a 2-digit number of bitmaps between 1 and 32 bit color depth, including several RLE formats. Here, adjacent, identical color codes are combined, so that the corresponding color and the associated number of occurrences

only have to be stored once. Particularly worth mentioning here is the RLE alpha format, which makes it possible to display one and the same image with different colors. Even better, changing the color does not require writing one line of code. In addition, each individual image can be swapped out to an external medium (e.g. SD card, SPI flash) so that it does not occupy any space in the addressable ROM area.

In addition, each individual font can be swapped out. You also have several options to save a lot of board space: First, AppWizard can limit the font to only those characters that actually need to be displayed on the screen. For example, suppose you only have a button that says »Example«, then instead of the entire alphabet, only the characters »E«, »x«, »a«, »m«, »p«, »l« and »e« would be included in the font. You can also save all the characters you want to use in a text file called a pattern file. Or simply specify the Unicode range you want to use.

And in the project dialog there are even more options, like bi-directional text or show missing characters – which is important for debugging, so you can see at a glance if characters are missing in your fonts. Last but not least, there is also support for complex font patterns like Thai or Arabic.

You mentioned maximum flexibility. When I create graphics in Excel, for example, I can base them on changeable values or animate them. Will something like this be possible in the future on the display of a refrigerator, for example, to show the fill level in the form of pie charts?

If the manufacturers think something like this makes sense, yes! Let me give you a simple example of what can be done at runtime using variables and triggers. Take a smartphone with two buttons on the case that allow you to change the volume. When you press a button, the current volume is displayed on the screen in the form of a slider whose length is proportional to the volume.



Image 1 shows the surface of AppWizard.
© Segger

When you reach the minimum volume, it triggers a trigger, over which a crossed-out speaker icon also appears. Thanks to the AppWizard, not a single line of code is required for such functions.

A few words about animations: An image widget is something every GUI builder has and is nothing particularly worth mentioning. Our image widget can set itself apart by the fact that with a few clicks you can create an animation that can scale and rotate any image at runtime. In the AppWizard you can find a small example of this with the WeatherForecast demo, where several small icons fly into the image from outside the screen. You can specify at which position they should stop, and how long the process should last, i.e. how fast they appear in the image.

However, that doesn't answer my question about the pie chart....

That's right, that's where we get to the Drawings. You just have to think of it as drawing functions for different elements like lines, circles, rectangles, and in our case, pie pieces. You specify the center of the circle, the radius, the start angle, the end angle, the color and off you go. It is exciting that the values can also be read from variables, for example, take a potentiometer as a volume control, whose position is graphically represented as a smaller or larger pie piece. For this purpose, it is possible to specify both a predraw and a postdraw for each individual object.

Take a button on which you want to draw a logo after it is displayed (Postdraw) or which should appear on a colored rectangle (PreDraw) – the options are limited only by the GUI designer's imagination. Among the samples in the AppWizard (File -> Open Sample) you can also find the PieChart sample.

The display is one side of the coin, the user interactions, where reactions to inputs occur, is another. What is special about your software?

Interactions bring life to the application and determine what has to happen and when. For this purpose, the so-called Interaction Window is available directly under the editor. This gives you an overview of all the interactions on a screen at any time, without having to switch to dialogs for individual objects.



Image 2 shows a temperature station, which was created by emWin and the AppWizard tool, without programming.
© Segger

A special feature here are conditional interactions. So you can define under which conditions, they are checked at runtime or an interaction is to be executed. In our example »How-toSetAlphaBitmap«, a magic hat is displayed together with three sliders that allow the user to change the color of the hat. In that case, the condition for a specific color is that the user sets the sliders to specific values. For functions with an »if-then« character, you do not need to program a line of code with the AppWizard.

Why did Renesas license emWin and the AppWizard?

Not only Renesas, but also other semiconductor manufacturers have been using our software for a long time, including STMicroelectronics, NXP, Infineon, Microchip, Dialog Semiconductor, Silicon Labs, Nuvoton and even Arms subsidiary Keil. The reason is very simple: with AppWizard and emWin you can implement high-end user guidance on touch displays without having to program a line of code. At the end, however, I don't want to hide one particularly exciting application: The team of Forze Hydrogen Racing has developed a steering wheel with display for their latest race car, they also rely on emWin and AppWizard.

How can I quickly find more information about emWin and AppWizard?

On our product page and in various videos, I would also like to recommend a video of the roundtable at Embedded World 2021.

Mr. Ehrle, thank you very much for your time.