

Design Secrets for Better HMIs Revealed

By Gavin Moore, Bridgetek

Since their very inception, car and motorcycle dashboards have mainly relied on mechanical, analogue dials (for speedometers, RPM counters, fuel gauges, etc.). There have, over last couple of decades, been minor additions (such as small LCD displays or indicator LEDs), but dashboards have still tended to basically follow the same analogue style. It is only in more recent times that real transformations have started to occur.

Technological advances started to be witnessed around five years ago that have brought about dramatic changes in the way vehicle information is displayed and control functions are executed. A new generation of human machine interfaces (HMIs) has thus emerged that are far more compelling and offer enhanced user experiences. There is now a great deal of design activity relating to this underway, particularly within the rapidly growing electric vehicle (EV) space - where market pressures mean that the need for differentiation is at its most acute.



Figure 1: The layout of a traditional automobile dashboard

The Evolution of Vehicle HMIs

The adding of features like GPS navigation technology proved to be an early game changer. This eliminated the need for drivers to refer to paper maps, which were often difficult to read (and obviously didn't offer auto updates), leading to far greater convenience. With GPS navigation, the step-by-step turn direction may be laid out clearly on the display. This makes it very easy to follow a route and estimate the time

of arrival at the selected destination.

With commuters spending longer periods inside their vehicles, the value of entertainment and communication have both been heightened too. Occupants are looking to gain access to a greater breadth of features in this respect (such as instant messaging with friends and family, streaming music, etc.) - and want the same sort of intuitive operation that their consumer gadgets (smartphones, tablets, etc.) deliver.

The advent of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technology, and the promise this brings of 'connected cars' on our roads, is set to have a strong influence on how future vehicle HMIs are laid out and the functionality they will possess. This will provide drivers with real-time updates concerning traffic congestion, possible accidents ahead and suchlike, as well as information on weather conditions, etc. It will also provide location data relating to nearby charging stations (vital for EV drivers, since charging or battery swap facilities are still quite scarce). In addition, it will allow the position of other vehicles travelling together in a group (whether for commercial or recreational purposes) to be determined - so that it can be ensured that none of them gets lost or is delayed in some way. Other key elements will include remote assist service - where drivers can push a button in order to contact emergency road service operatives, as well as allowing real-time remote diagnostic capabilities to be benefitted from.

However, with so many features and functions that drivers need to access, it presents engineers with a considerable challenge to fit everything onto the dashboard - especially when there are space limitations to consider. Generally, all this navigation information and details of infotainment content will be displayed on the central console, separate from the speedometer and RPM dials. However, in a motorcycle/EV scooter context, since there is only one main display available, it is necessary for such data to share the screen with key operational figures. In some cases, it may be possible to switch between different screens at regular intervals.

It isn't just about available space though. It should also be noted that that the rendering of too much information at once may be distracting for the driver - potentially hindering them by taking their focus off the road, and increasing the risk of an accident occurring. This is especially true for motorcyclists and EV scooter users, who need to devote as much attention as possible to the environment around

them (other road users, the road surface conditions, etc.).

Migration to Digital Dashboards

Though we have focused solely on this so far, it should be made clear that going digital is not only advantageous from the driver's perspective - vehicle manufacturers want to push forward with this too. Firstly, the removal of mechanical dials from the system means there are less moving parts - that might stop working and need to be replaced by a servicing engineer at some stage (and are typically more expensive than a display too). It also means that the system can be more easily upgraded in the future so that new features are added, or the aesthetics improved.

So LCD displays can be used for imparting accurate information, like simple digits, without the need for cutting-edge graphics ICs. To greatly improve the look of the display, and enhance the user experience derived from it, high performance graphic acceleration is normally going to be required though - and this brings a whole series of engineering problems with it.

The implementation of highly detailed and complex displays, which are rich in functionality and constantly refreshed (such as the one shown in Figure 2) using conventional technology would call for hardware that is expensive, power consuming and occupies a lot of board space. The main reason for this being that each pixel is rendered separately - and this calls for a large amount of processing capacity and memory resource.



Figure 2: Modern EV dashboard with numerous essential parameters shown on the LCD display

Bridgetek’s Embedded Video Engine (EVE) technology offers a way to circumvent this, by taking a unique object-oriented approach. Here all the elements (dials, gauges, graphs, etc.) within the HMI are allocated a sort of ‘shorthand’ reference means that they can easily be taken from the display list and don’t have to keep being redrawn. This enables the construction of the multi-faceted HMIs that the market now expects, but while keeping the associated bill-of-materials (BoM) costs to a minimum and without requiring heavy investment of time or engineering expertise.

The EVE graphic controller device can be connected to a low power, low cost microcontroller. This arrangement frees up the processing capacity of the microcontroller so it can carry out its main tasks without lag. In the Figure 2 example, the dashboard has been rendered using an FT813 EVE device, running at a speed of less than 100MHz with no external memory attached. This has been paired with a relatively inexpensive 32-bit microcontroller (the STM32F4 from STMicroelectronics), plus an 800x480 pixel resolution LCD display. It all runs smoothly at a 60fps frame rate, with only 64kB of memory being required. Just 75% of the EVE’s graphical resources are needed to get results of this quality.

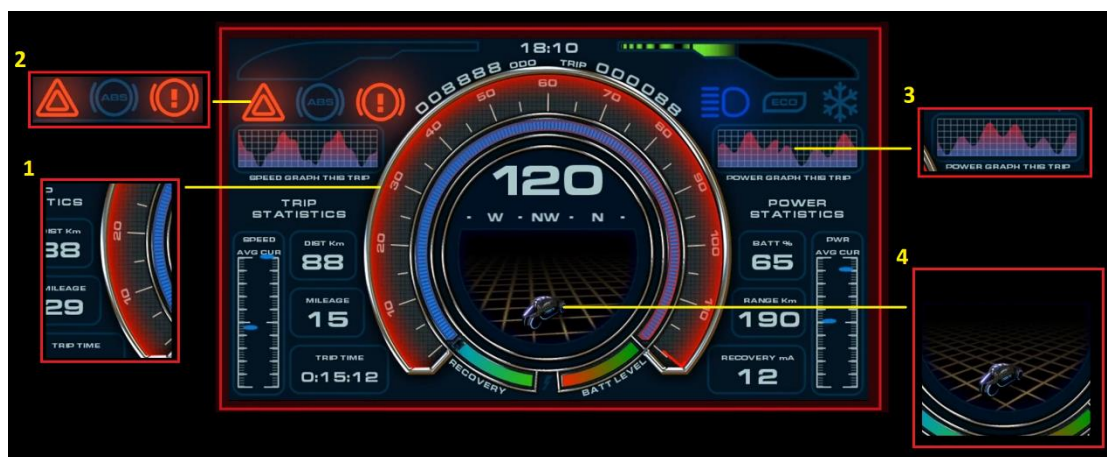


Figure 3: More detailed description of the dashboard functions

Let’s now look at some specific areas of the EV dashboard example. Figure 3 has several items marked out on it. The most important feature of any vehicle dashboard will be the speedometer (denoted as item 1). Despite that fact that they want the digital displays, drivers still like the idea of speed (or RPM) being signified via needles in the round dial format that, in some cases, they have become familiar with over many decades of vehicle ownership.

The needle needs to point correctly at the speed figure, so the graphical content has to be delivered with high enough resolution to provide sufficient accuracy. They must also be refreshed rapidly enough that the dial does not lag behind and acceleration of the vehicle. EVE's dial widgets simplify the creation of features like these, substantially reducing the lines of code involved. As a consequence, the EVE graphic controller can take care of rendering graphics onto the display, so the microcontroller doesn't have to get involved in the processing work. Thanks to EVE's object-based architecture and use of its stencil function, only the relevant section of the dial needs to be updated (rather than having to constantly refresh the entire screen). This streamlines the system and means that there can be less constituent hardware involved.

For the various indicators that are situated around the periphery of the central dial (e.g. lights, driving mode, etc.), it is possible to create an eye-catching glowing effect (shown in item 2 of Figure 3). This is achieved through use of the hardware alpha-blending function. The indicators are each saved in an L8 greyscale bit format. Then bitmap transformations are employed (for rotation, scaling and blending them with special color bitmaps) to render the final indicators. The code used to create this effect is shown below.

```
//Specify different drawing colors depending on the user control
if (color_switch_timer > 128)
    App_WrCoCmd_Buffer(phost, COLOR_RGB(0x35, 0xEA, 0x19));
else
    App_WrCoCmd_Buffer(phost, COLOR_RGB(0x05, 0x15, 0x22));
App_WrCoCmd_Buffer(phost, BEGIN(BITMAPS));

//Specify the way to blend source color with destination color, see OpenGL
API glBlendFunc for details.
App_WrCoCmd_Buffer(phost, BLEND_FUNC(SRC_ALPHA, ONE));

//Draw icon image "ABS"
App_WrCoCmd_Buffer(phost, VERTEX2II(64, 190, ABS_ICON_HANDLE, 0));

//Draw glow image "A"
App_WrCoCmd_Buffer(phost, VERTEX2II(392, 116, GLOWIMAGE_HANDLE, 0));
```

Also in the example shown, the speed graph and power graph (item 3 in Figure 3) give the impression of being quite complex, but are not actually difficult to create - it is just a matter of combining several different visual components together. Here the L4 format bitmap is used to form the grid background. Then alpha-blending of the speed histogram with gradient color (going from red at the top to blue at the bottom) completes the effect.

Another feature that is starting to become popular in HMIs is inclusion of animated content. This is true not just in vehicles, of course, the trend is also being witnessed in the retail (for point-of-sales units), hospitality (hotel lifts, etc.) and industrial automation sectors (where it can be used to help staff in replacing parts through the playing of an instructional animation). In the example there is a 3D bike toward the bottom of the central dial (see item 4 in Figure 3), which provides a way to indicate the orientation of the actual vehicle. This has been created by the compiling together of multiple images, covering every angle of the rendered bike. These are saved in one large bitmap file - rather than as individual files, which would take up much more memory space.

As the proliferation of EVs continues, the need to offer more satisfying user experiences will become increasingly apparent. Engineers will need to accomplish this while at the same not adding to the system cost or the power budget. Rather than just throwing processing resource at it, a more sophisticated and well-thought out tactic is required - one that fully respects the various constraints involved.