

HMI-GERÄTE FÜR OEM-APPLIKATIONEN SELBST KONFIGURIEREN

Individuelle Schnittstelle

Jedes komplexere technische Gerät benötigt eine Schnittstelle zum Bediener. Das Angebot an solchen Human Machine Interfaces (HMIs) ist schier unüberschaubar geworden. Der hier vorgestellte Ansatz zum Selbstgestalten führt in wenigen Schritten zum Instrument der Wahl.

KONTAKT

GeBE Elektronik und Feinwerktechnik GmbH,
82110 Germering,
Tel. 089 894141-31,
Fax 089 894141-33,
www.oem-printer.com

KLAUS BALDIG

Zu den grundlegenden Aufgaben eines HMI-Geräts gehört es, sowohl die Eingabe als auch die Ausgabe von Daten zu verarbeiten. Diese können etwa über analoge oder digitale Eingänge,

USB, CAN oder Ethernet zum HMI-Gerät gelangen und von dort wieder versendet werden. Das Gerät sollte außerdem in der Lage sein, abhängig von vordefinierten Bedingungen Daten zu lenken oder definitionsgemäß darauf zu reagieren.

Was der Markt zu bieten hat

Bewegt man sich etwas tiefer in den Markt der HMI-Geräte, stellt sich die Situation für OEM-Entscheider wie folgt dar: Einerseits wird eine Vielzahl an Geräten angeboten, die im Prinzip kleine EinbaupCs sind, meist mit Windows CE ausgerüstet. Andererseits findet man Geräte,

deren Tools es dem Benutzer einfach machen, seine eigene Applikation ohne Programmierkenntnisse zu erstellen. Meist sind diese jedoch auf ein bestimmtes, branchenspezifisches Aufgabengebiet ausgerichtet und daraufhin optimiert. Das erschwert dem branchenfremden Benutzer den Einsatz zu seinem Zweck oft erheblich. Oder die Tools bieten zugunsten einfacherer Bedienung nur grafische Standardelemente, die höheren grafischen Ansprüchen nicht genügen und wenig ansprechend wirken.

Wer sich bei seinem HMI mit der Darstellung eines Windows-CE-Desktop-



WISSENSWERT

Die grafische Darstellung. Prinzipiell sollten weiterhin grafische Standardelemente beim Erstellen der HMI-Oberfläche verwendet werden, denn das hat den Vorteil, dass das Design schnell, einfach und ohne tiefe Programmierkenntnisse umgesetzt werden kann. Der Nachteil eingeschränkter Designvielfalt kann überwunden werden, indem diese Standardelemente zusätzlich frei editierbar gemacht werden. Desktop-Publishing-Programme arbeiten beispielsweise nach dieser Philosophie. Jedes Element enthält Eigenschaften, die über ein Menü eine freie Gestaltung ermöglichen. Typische HMI Elemente sind beispielsweise Analoginstrumente, Bargrafen (auch Füllstandanzeigen), ein- oder mehrzeilige Textfelder, Tastenfelder zur Eingabe auf dem Touchscreen oder Hot-Key-Anzeigen für mechanische Taster außerhalb des Displays, Anzeigefelder wie virtuelle LEDs oder auch nur Grafiken, statische Texte und Logos. Hinzu kommen Darstellungen von bewegten Bildern wie Filmen oder Diashows.

Die Funktion der Darstellung. Ist das grafische Design erstellt, muss in den Eigenschaften des Elementes selbst festgelegt werden, welche Ein- oder Ausgangskanäle oder Aktionen es gibt.

Bei einem Textfeld bedeutet das zum Beispiel, ob die variablen Anzeigedaten von einer Schnittstelle kommen oder ob sie Messdaten eines analogen Messeingangs sind. Bei einem Tastenfeld stellt sich die Frage, wohin die Information des Tastendrucks gelangen soll.

An dieser Stelle wären die Möglichkeiten für einen transparenten Betrieb vermutlich ausreichend, da alle Verarbeitungsentelligenz im übergeordneten Host-System liegt. Doch oft steht ein solches Host-System gar nicht zur Verfügung. Wird das HMI beispielsweise als analoges Messgerät verwendet, wäre es wünschenswert, etwa einen Alarm auszugeben, wenn ein vordefinierter Messwert überschritten wird. Dafür benötigt es selbst zumindest eine einfache Intelligenz.



Computers zufrieden gibt, kann die Lektüre dieses Artikels hier beenden. Für jeden anderen Entscheider tun sich erst jetzt die wahren Probleme und Kosten auf. Denn der oben beschriebene kleine Einbau-PC benötigt selbstredend eine Software, die ihn erst zum HMI-Gerät werden lässt. Und diese entsteht im Regelfall unter den fachkundigen Händen eines Software-Entwicklers. Zwar gibt es mächtige Grafik-

bibliotheken wie „QT“, die ihm zur Seite stehen. Doch spezifisches Fachwissen ist unabdingbar, um ein ansprechendes HMI-Design zu entwickeln.

Ausgehend von dem kleinen Einbau-PC, dem nun der Software-Entwickler ein Gesicht geben muss, lässt sich ein typisches Alltagsbild zeichnen: Die Software-Oberflächen werden wieder und wieder überarbeitet, weil – trotz ausführlicher Spezifikation – der Programmierer die Gedanken und den Geschmack der Entscheider nicht kennen kann. Der gesamte Prozess verschlingt viel Zeit und viel Geld, bis letztlich, nach vielen Rekursionen, für den Entscheider (meist nur) ein Kompromiss übrig bleibt.

Die Frage heißt also: Wie lässt sich eine individuelle HMI-Lösung ohne Kompromisse (also möglichst vom OEM selbst) generieren, die auch noch in kleinen und mittleren Projekten finanziell realisierbar bleibt?

In wenigen Schritten zum Ergebnis

Bei der Entwicklung eines geeigneten Konzepts stehen vor allem zwei Aspekte im Fokus: die grafische Darstellung selbst sowie ihre Funktion (siehe **Ⓛ-Kasten**). Ein bei GeBE entwickeltes, PC-basiertes Designtool bildet zusammen mit der HMI-Geräteserie „Indico“ eine Lösung, die den schnellen Einsatz in unterschiedlichen Applikationen ermöglicht – und das ohne die Notwendigkeit, Software zu schreiben. Das Designprogramm

basiert grafisch darauf, dass Bilder (etwa ein Zeiger) vor einem definierten Hintergrund (wie einer Skala) bewegt werden. Der Anwender kann damit ein analoges Instrument grafisch komplett erzeugen oder aber einen fertigen Hintergrund und Zeiger verwenden. Das Programm lädt diese Bilder zusammen mit einer Anweisungsdatei in das HMI-Gerät. Diese Datei gibt vor, wie die Bilder zu verwenden sind, welche Ein- und Ausgangskanäle verwendet werden und welche Aktionen stattfinden sollen.

Das Programm bietet eine Vielzahl grundlegender Formen zur grafischen Darstellung an. Im Folgenden soll beispielhaft dargestellt werden, wie einfach darin ein analoges Instrument erstellt werden kann: mit einem analogen Eingangskanal und einem Alarm, der eine virtuelle LED aufleuchten lässt, sobald ein Messwert überschritten wird.

1. Welches Instrument zeigt an?

Zunächst muss für das neue HMI-Projekt ein passender Hintergrund gewählt werden. Es können beliebige Hintergründe, Farbverläufe, Bilder, Logos und Texte hinzugefügt und gestaltet werden. Im Weiteren kann man das gewünschte Rundinstrument auswählen. Mit dem Skalentyp (drehend, geradlinig, numerisch) legt man die grafische Fläche, die für die analoge Darstellung eines Messwertes benötigt wird, und die Art der Messwertdarstellung fest. Im Fall der Rundskala können Mittelpunkt, Anfangs- und Endwinkel angegeben werden. Je nach Element lässt sich die Darstellung in Farbe, Breite, Länge und Schrift gestalten. Der Vorgang ist mit dem Entwurf einer Siebdruckvorlage für ein Zifferblatt vergleichbar (**Bild 1**).

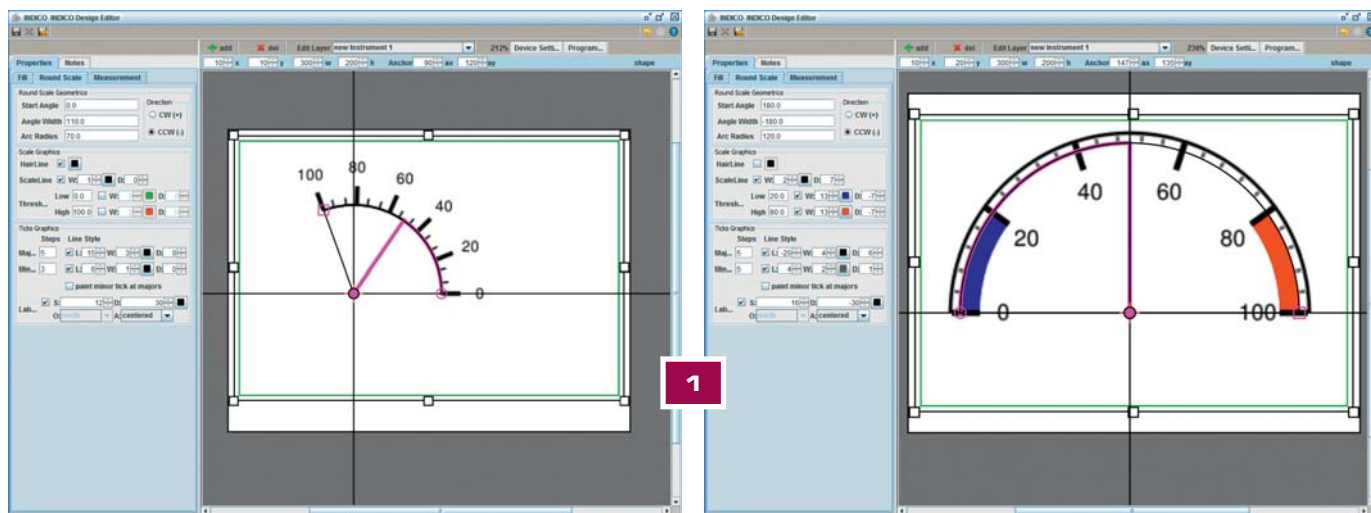


Bild 1. Festlegen des Instruments mit Skala und Zeiger

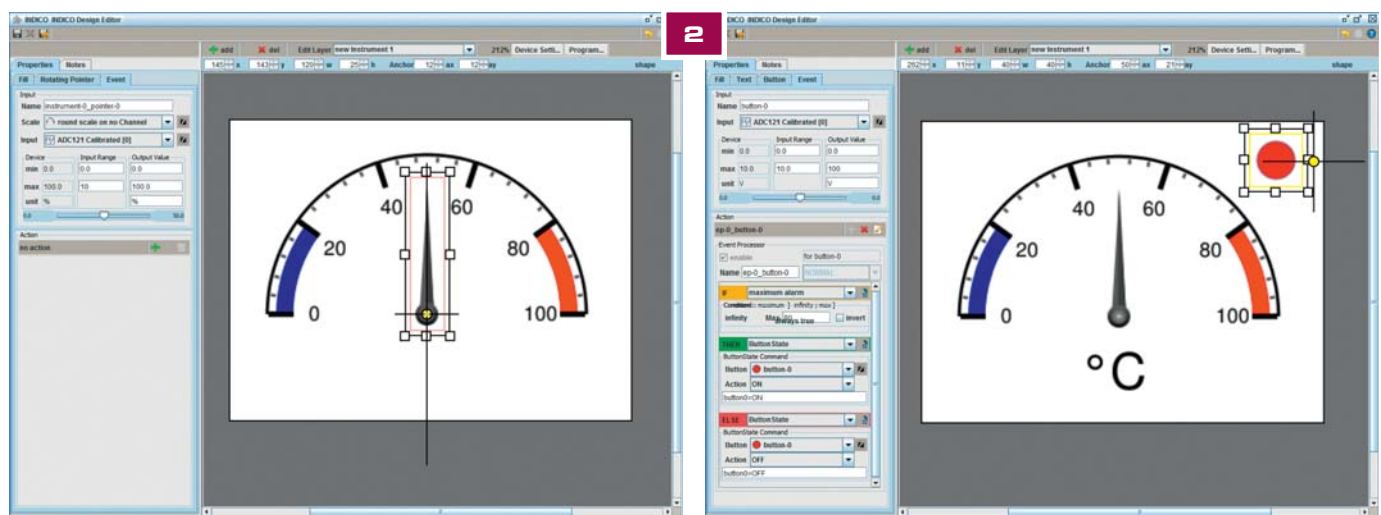


Bild 2. Das Umrechnen der RAW-Werte in die darzustellende Messgröße erledigt der Signalprozessor (links); der Event-Prozessor steuert die Reaktionen auf Systemkommandos (rechts)

2. Mit welchem Zeiger wird angezeigt? Der Designer bietet unterschiedliche editierbare grafische Darstellungen von Zeigern (Pointer); außerdem lassen sich Bilddateien importieren. Der interessante Aspekt eines Zeigers ist die Zuordnung eines Messwerts zu einem grafischen Messbereich, wie er in einer Skala definiert wird. Nachdem ein Pointer erstellt und seine Eigenschaften gestaltet sind, muss er noch auf eine Skala gesetzt werden – ähnlich dem Aufstecken des Sekundenzeigers auf die mittlere Achse einer Uhr. Damit wird festgelegt, wie er seinen Messwert darstellen soll, wo der Nullpunkt, wo das Minimum und wo das Maximum ist, ob er sich um den Mittelpunkt

der Skala von Winkel bis Winkel drehen oder als eine Linie vom Nullpunkt zum anzuzeigenden Wert (im Messwertbereich) erscheinen soll. Numerische Zeiger besitzen zwar keine grafische Animation, dafür aber eine Schriftgröße, eine Farbe und eine Formatierung (Bild 2, links).

3. Welcher Messwert wird dargestellt? Jedem Zeiger wird ein Geräte-Input-Kanal zugewiesen, aus dem er die Werte erhält, die er anzeigen soll. Das können analoge oder auch Dateneingänge wie RS232, USB, CAN oder Modbus sein. Da es sich oft um RAW-Werte handelt, kann im Designprogramm festgelegt werden, wie diese in die darzustellende Messgröße umzurechnen sind. Bild 2

zeigt rechts einen analogen Input von 0 bis 10 V, der einer Geschwindigkeit von 0 bis 100 °C entspricht. Wie in diesem Beispiel können einfache lineare Skalierungen im Gerät programmiert werden.

4. Wie und womit wird der Alarm ausgelöst? So wie alle Zeiger und Taster besitzt ein Instrumentendesign Beobachtungsfunktionen: Ein Event-Prozessor stellt eine Fragebedingung an den Messwert des beobachteten Kanals (wie: größer als dieser Wert) und programmiert das HM-Gerät, im Falle einer Übereinstimmung eine Folge von Systemkommandos auszuführen, etwa einen Alarm auszugeben. Für unser Beispiel muss noch ein Action Button definiert werden, der einen Event-Prozessor zugewiesen bekommt. Im Event-Prozessor wird das Ereignis Messwert > 80 mit der gewünschten Aktion verbunden, nämlich die virtuelle LED rot leuchten zu lassen (Bild 2, rechts).

5. Fertig zum Einsatz. Per Download-Aktion wird dieses Projekt jetzt noch in das HMI-Gerät geladen, und das Messgerät ist nach fünf Schritten einsatzbereit. (m/)



FAZIT

Für OEMs einfach und Kosten sparend. Die Erläuterung der wenigen Schritte zu einem hier zwar einfachen, aber eben genau nach Wunsch angelegten, funktionsfähigen OEM-Instrument macht es deutlich: Das hier vorgestellte Designprogramm ist auch ohne ein technisches Studium problemlos erlern- und anwendbar. Damit erlaubt es dem OEM den betriebswirtschaftlich sinnvollen Einsatz moderner grafischer HMI-Technik – auch in kleinen und mittleren Projekten. Der Funktionsumfang des Programms ist weitaus größer als im gezeigten Beispiel, sodass Kunden damit auch andere Applikationen wie analoge oder digitale Messgeräte, Parkschraken-Anlagen, Automaten, Fahrgast-Displays, Geräte der Gebäudeautomation oder einfach nur ein Textdisplay-Ersatz ohne zusätzlichen Eingriff in die Geräte-Firmware selbst realisieren können.



DER AUTOR

Dipl.-Ing. (FH) **KLAUS BALDIG** ist Leiter Entwicklung bei GeBE in Germering bei München.

www.EL-info.de

7 156 02