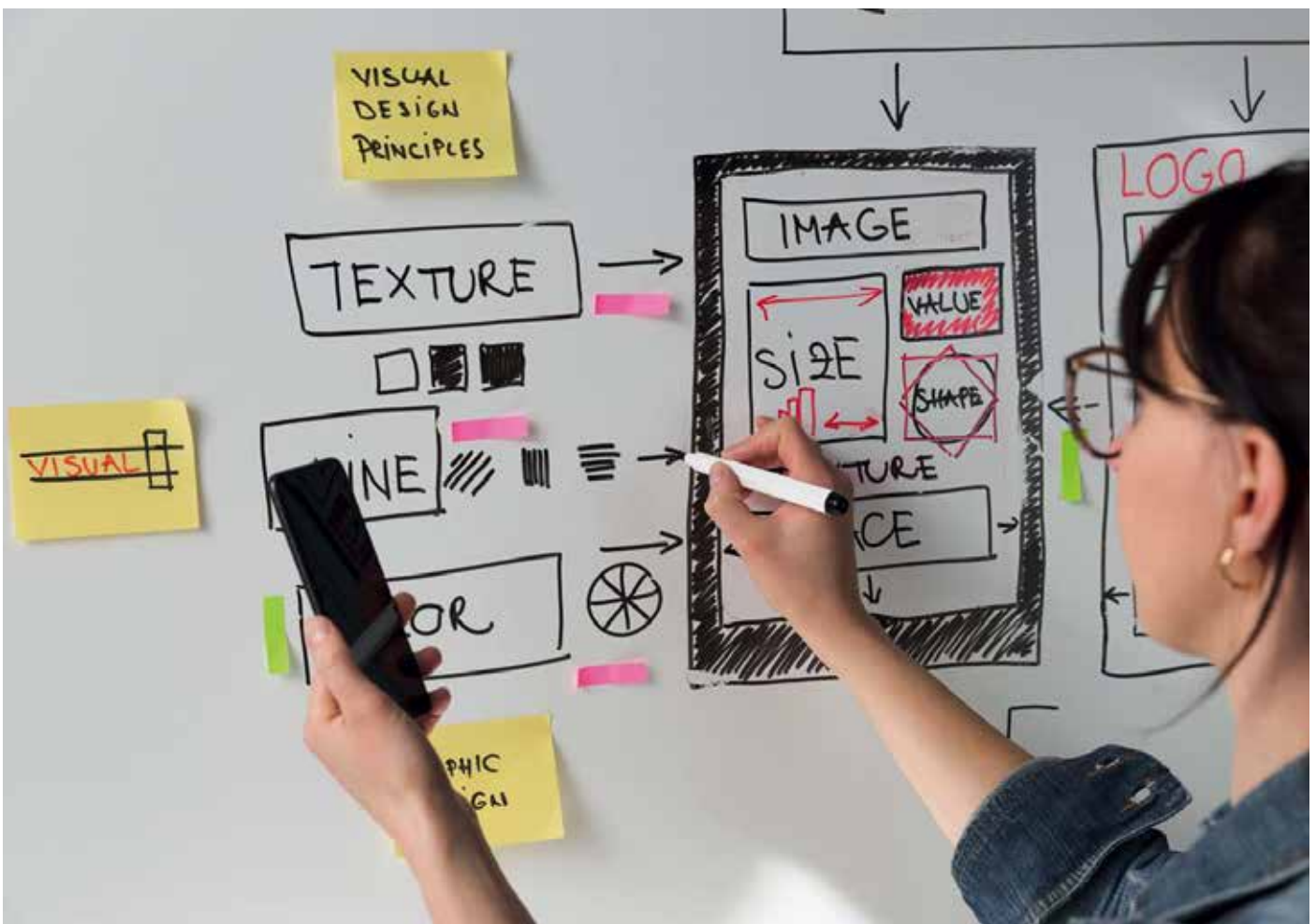


Elektronik

Benutzeroberflächen für Embedded-Systeme

Vorbild Smartphone

Grafische Benutzeroberflächen (GUIs) auf Embedded-Systemen mit Touchdisplays müssen auf ressourcenbeschränkten Mikrocontrollern laufen können. Dennoch sollen sie dem Smartphone-Vorbild entsprechen. Mit einer ausgeklügelten Grafikbibliothek und dem geeigneten Designtool lassen sich moderne GUIs ohne großen Aufwand erstellen. Von Axel Wolf



LCD-Controller und Grafikprozessoren auf Mikrocontrollern sowie die dazugehörige Software gewinnen in der Embedded-Industrie immer mehr an Bedeutung. Der Grund dafür ist recht einfach: Die Kunden von heute sind verwöhnt. Warum? Weil jeder von uns ein Smartphone in der Tasche hat. Und wir alle lieben es, damit

umzugehen. Deshalb verbringen wir so viel Zeit damit. Das bedeutet, dass speziell jüngere Kunden die gleiche – oder zumindest eine sehr ähnliche – Benutzererfahrung auch auf anderen elektronischen Geräten erwarten, sei es ein Thermostat, eine Kaffeemaschine, ein Blutzuckermessgerät oder ein Backofen.

Um eine Smartphone-ähnliche grafische Benutzeroberfläche auf solchen Geräten zu implementieren, müssen zwei große Herausforderungen bewältigt werden:

→ Das erste Problem ist die Rechenleistung. Embedded-Anwendungen laufen meistens auf einem Prozessor der ARM-Cortex-M-Klasse, der nicht mit

der Leistung der Prozessoren in einem Mobiltelefon mithalten kann. Die zweite Herausforderung ist der Speicher, der offensichtlich auf einem Embedded-System viel knapper bemessen ist als auf einem Smartphone. Und es gibt sogar eine dritte Herausforderung, die nicht zu unterschätzen ist und die Arbeitsabläufe und Kompetenzen der Entwicklungsteams betrifft. Auf der einen Seite gibt es die Softwareentwickler, die wunderbar in einer Programmiersprache wie C programmieren können, aber nicht unbedingt ein Auge für das GUI-Design haben. Und auf der anderen Seite gibt es die Grafiker, die die schönsten Benutzeroberflächen im Sinn haben, aber meist nicht wissen, wie man programmiert. Infolgedessen gibt es oft zu viele Abstimmungsmeetings zwischen zwei Gruppen von Menschen, die auf völlig unterschiedliche Weise kommunizieren – den kreativen Grafikdesignern und den eingefleischten Softwareentwicklern. Um diese Herausforderungen zu bewältigen, wird eine Embedded-Grafikbibliothek benötigt, die in Bezug auf die Speichernutzung äußerst effizient ist, aber dennoch die Leistung bietet, die für Smartphone-ähnliche GUIs erforderlich ist. Außerdem benötigen die Grafiker ein Entwicklungswerkzeug, mit dem sie diese GUI vollständig visuell erstellen können, ohne programmieren zu müssen. Nachdem die Benutzeroberfläche erstellt wurde, kann sie dann an die Softwareentwickler weitergegeben werden, um sie in die Anwendung zu integrieren. Eine Antwort auf diese Herausforderungen ist die Grafikbibliothek emWin von Segger Microcontroller und das passende GUI-Design-Tool AppWizard, die beide in diesem Beitrag vorgestellt werden.

Anwendungsbeispiel: Eine Smartphone-ähnliche Backofensteuerung

Um die Leistungsfähigkeit von emWin und die Bedienbarkeit des AppWizards in der Praxis zu demonstrieren, wurde das Ziel gesetzt, eine Touch-gesteuerte Benutzeroberfläche für einen Backofen mit Icons und Bedienelementen zu entwickeln, wie sie von Smartphones bekannt

sind. Heute gibt es zwar schon Backöfen mit Touch-Oberfläche, aber diese enthalten meist Listenfelder mit Textinhalten, und die Auswahl erfolgt nicht direkt über Icons, sondern indirekt über Scrollbalken und dedizierte Buttons.

Die Funktion dieser Demo-Anwendung ist im Vergleich zu handelsüblichen Backöfen etwas eingeschränkt, zeigt aber alle wichtigen Funktionen wie Betriebsart, vordefinierte Rezepte, Temperatur, Backzeit und ist sogar mehrsprachig implementiert. **Bild 1** zeigt die implementierte Benutzeroberfläche vom Start bis zum Beginn des Backvorgangs.

Grafikbibliothek emWin für Embedded-Systeme

Um eine Smartphone-ähnliche Benutzeroberfläche zu implementieren, ist eine Embedded-Grafikbibliothek mit außergewöhnlichen Eigenschaften erforderlich. Die wichtigsten werden im Folgenden vorgestellt.

Flexible Einrichtung

Entwickelt wurde emWin für effiziente Prozessor- und Display-Controllerunabhängige grafische Benutzeroberflächen für alle Anwendungen, die ein grafisches Display verwenden. Sie ist kompatibel zu Singletask- und Multitask-Umgebungen, mit proprietären oder kommerziellen Betriebssystemen und kann mit jedem Display-Controller und jeder CPU an jede Größe physikalischer oder virtueller Displays angepasst werden. Immer häufiger stellen Mikrocontroller Hardwarebeschleunigung für grafische Operationen zur Verfügung. Sie entlasten die CPU für alle oder zumindest die meisten grafischen Operationen, indem sie DMA-Engines oder andere Hardwareblöcke verwenden, die hohe Datendurchsätze ermöglichen. So unterstützt emWin beispielsweise Hardwarebeschleuniger wie Chrom-ART von STMicroelectronics oder die 2D Drawing Engine (DRW) von Renesas.



Bild 1. Benutzeroberfläche der Demo für eine Backofensteuerung auf einen Blick. Einige Screens sind der Einfachheit halber nicht dargestellt. (Bild: Segger Microcontroller)

Unterstützung für mehrere Ebenen

Viele moderne LCD-Controller bieten die Möglichkeit, mehrere Anzeigeebenen darzustellen. In emWin kann jede Ebene ihre eigene Farbtiefe, Größe und Position haben. Jede Ebene kann sogar falls notwendig seinen eigenen Displaytreiber haben. Dadurch ist es möglich, mehrere Displays gleichzeitig anzusteuern oder sowohl mehrere Ebenen also auch mehrere Displays zu verwenden.

Unterstützung zahlreicher Displaytreiber

Die Grafikkibliothek emWin unterstützt zahlreiche Displaytreiber und somit auch jedes gängige Display, z. B. Schwarzweiß-, Graustufen- und Farbdisplays. Sie deckt eine breite Palette von Anwendungen ab, die von medizinischen Geräten wie Herzfrequenzmessern bis hin zu komplexen Infotainment-Displays reichen.

Ressourceneffizienz

Dank der hohen Ressourceneffizienz von emWin reichen 130 KB RAM und 256 KB Flash-Speicher oft aus, um eine benutzerfreundliche und moderne Anwendung zu entwickeln, einschließlich Fenstermanager und Widgets (siehe Abschnitt Widgets und Window Manager). Darüber hinaus bietet emWin verschiedene Einstellungen, um den RAM- und ROM-Bedarf weiter zu optimieren, beispielsweise indem der Fenstermanager ohne transparente Fenster verwendet oder die Textrotation deaktiviert wird.

Unterstützung verschiedener Schriftarten

Die Darstellung von Text in emWin basiert auf der UTF-8-Codierung. Die Grafikkibliothek unterstützt das Zeichnen aller Zeichen der Basic Multilingual Plane (BMP, Ebene 0), die die Codepunkte von 0x0000 bis 0xFFFF abdeckt. Darüber hinaus können bidirektionale Skripte und Texte verwendet werden. Schriftarten, die als C-Font-Dateien, als TrueType-Font-Dateien, als SIF, XBF oder iType implementiert werden können, unterstützt emWin. Um Speicherplatz zu sparen, bietet emWin drei Möglichkeiten:

- Die erste ist eine Beschränkung des Zeichensatzes auf nur die tatsächlich auf dem Display angezeigten Zeichen.
- Der GUI-Entwickler kann zusätzlich auch alle Codepunkte, also die in der Anwendung verwendeten Zeichen, in einer Pattern-Datei speichern.
- Nicht zuletzt kann auch der verwendete Unicode-Bereich angegeben werden.

Unterstützung einer breiten Palette von Bitmaps

Neben BMP-, JPEG-, GIF- und PNG-Dateien unterstützt emWin auch mehrere emWin-spezifische Formate. Der empfohlene und effizienteste Weg, eine zur Kompilierzeit bekannte Bitmap anzuzeigen, ist die Verwendung eines der emWin-spezifischen Formate. Wenn die Anwendung zur Kompilierzeit unbekannte Bilder anzeigen muss, muss das Bild in einem von em-

Win unterstützten Grafikdateiformat verfügbar sein. In diesem Fall kann sich die Bilddatei im Arbeitsspeicher oder einem anderen Speichermedium befinden. Es kann auch dann angezeigt werden, wenn der verfügbare Speicherplatz geringer ist als die Größe der Bilddatei. Nativ unterstützt emWin BMP-, JPEG- und GIF-Dateien; PNG-Dateiunterstützung kann erreicht werden, indem die PNG-Bibliothek hinzugefügt wird, die auf der Segger-Website verfügbar ist.

Mehr als ein Dutzend Bitmap-Formate zwischen 1 und 32 bit Farbtiefe bietet emWin, darunter mehrere Run-Length-Encoding-Formate (RLE). Identische benachbarte Farbcodes werden kombiniert, sodass die entsprechende Farbe und die entsprechende Häufigkeit nur einmal gespeichert werden müssen. Das RLE-Alpha-Format ermöglicht es, dasselbe Bild mit unterschiedlichen Farben darzustellen. Alle Icons in der Backofen-App sowie die Bildschirmhintergründe sind als emWin-spezifische Bitmaps implementiert, um Speicherplatz zu sparen (**Bild 2**).

Außerdem kann jedes einzelne Image auch auf ein externes Medium (SDKarte, SPI-Flash etc.) ausgelagert werden, sodass es keinen Platz im adressierbaren ROM-Bereich belegt. Jede einzelne Schriftart kann ebenfalls ausgelagert werden.

Widgets und Window Manager

Eine Reihe von Routinen zum einfachen Erstellen, Verschieben, Vergrößern/Verkleinern und Bearbeiten beliebig vieler Fenster wird vom emWin PRO Window Manager (WM) bereitgestellt. Der erlaubt die gleichzeitige Verwendung sowie das Übereinanderschichten verschiedener Fenster. Als Steuerelemente dienen emWin-Widgets, das sind Fenster mit objektartigen Eigenschaften. Sie reagieren auf bestimmte Ereignisse, ermöglichen eine Benutzerinteraktion und ihre Erscheinung kann in den meisten Fällen einfach angepasst werden.

Das Backofenbeispiel verwendet unter anderem das »Rotary«-Widget zum Einstellen der Temperatur und das »Switch«-Widget zum



Bild 2. Hintergründe und Icons werden unter Verwendung von emWin-spezifischen Bitmap-Formaten implementiert, um einen möglichst geringen Speicherbedarf zu gewährleisten. (Bild: Segger Microcontroller)



Bild 3. Derzeit im AppWizard verfügbare Widgets. (Bild: Segger Microcontroller)

Auswählen eines Eco-Modus. **Bild 3** zeigt alle derzeit im AppWizard verfügbaren Widgets. Neben den vordefinierten Widgets können auch Zeichenfunktionen für verschiedene Elemente wie Linien, Kreise, Rechtecke oder Kreissegmente verwendet werden. Um ein Kreissegment zu zeichnen, kann einfach der Mittelpunkt des Kreises, der Radius, der Start- und Endwinkel und die Farbe angegeben werden (**Bild 4**). Das Geniale ist, dass diese Werte auch aus Variablen ausgelesen werden können, sodass sich beispielsweise die aktuelle Position eines Schiebereglers grafisch als kleineres oder größeres Kreissegment darstellen lässt.

Es ist möglich, für jedes einzelne Objekt sowohl eine Vorzeichnung als auch eine Nachzeichnung festzulegen. Dies sind Aktionen, die vor oder nach der Anzeige des Elements ausgeführt

werden. Beispielsweise könnte eine Nachzeichnung verwendet werden, um ein Logo auf einer Schaltfläche darzustellen, nachdem die Funktion ausgeführt wurde. Mit einer Vorzeichnung könnte die Schaltfläche auf einem farbigen Rechteck angezeigt werden. Die Möglichkeiten sind nur durch die Phantasie des GUI-Designers begrenzt.

Unterstützung mehrsprachiger Anwendungen

Obwohl sich die Ausgabe von Text in den meisten Sprachen auf die einfache Ausgabe der gewünschten Zeichenketten reduziert, erfordert die Ausgabe von arabischen, hebräischen, thailändischen oder bidirektionalen Texten etwas mehr Aufwand. Um die Sprachunterstützung einzuschalten, sind beim Starten der Anwendung nur wenige Funktionsaufrufe erforderlich, und alle textbasierten Funktionen unterstützen automatisch die erforderlichen Sprachen. Die Text- und Sprachressourcendateien erleichtern die Erstellung mehrsprachiger Anwendungen. Der GUI-Entwickler kann eine oder mehrere Text- oder CSV-Dateien speichern, die Text in verschiedenen Sprachen enthalten. Dies ermöglicht es, den Text einer Anwendung zu ändern, ohne den Code ändern zu müssen. Im Backofenbeispiel ist unter anderem eine Sprachauswahl für chinesische und griechische Schriftzeichen (**Bild 5**) implementiert. Auch andere Sprachen wie Thai oder Arabisch lassen sich so einfach realisieren.

Integration von Benutzerinteraktionen und Animationen

In unserer Beispielanwendung werden viele Benutzerinteraktionen verwendet, was kaum verwundert, da jeder Tipp auf ein Icon oder Steuerelement eine entsprechende Reaktion auslösen muss. Dennoch sollen zwei Beispiele, die in der Anwendung auffallen, näher beschrieben werden: Die Temperatureinstellung und der Start des Backvorgangs. Die Backofentemperatur wird über ein Widget vom Typ »Rotary« eingestellt. Neben dem Drehknopf

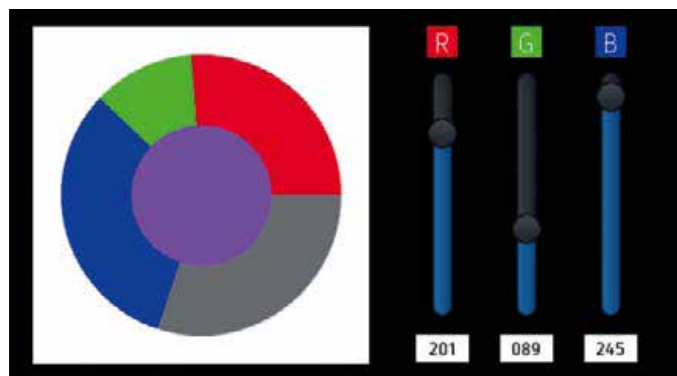


Bild 4. Erstellen eines Kreisdiagramms mit Hilfe von Zeichnungen. (Bild: Segger Microcontroller)



Bild 5. Sprachauswahl in der Backofen-Beispielanwendung. (Bild: Segger Microcontroller)



Bild 6. Die Farbe des angezeigten Textes entspricht der aktuell durch den Drehknopf ausgewählten Temperatur. (Bild: Segger Microcontroller)

wird die aktuell eingestellte Temperatur als Zahl angezeigt. Das Bemerkenswerte daran ist, dass AppWizard-Interaktionen die Textfarbe je nach Absolutwert der Temperatur von Blau über Pink nach Rot ändern – und das in Echtzeit, während der Nutzer am Drehknopf dreht (**Bild 6**). Später, wenn der Backvorgang startet, wird zunächst der Backofen vorgeheizt. Dazu gibt es einen Fortschrittsbalken und ein Textdisplay, das die verbleibende Aufheizzeit anzeigt und bis auf null herunterzählt. Wenn der Backofen vorgeheizt ist, wechselt der Fortschrittsbalken zu einer Anzeige für die verbleibende Backzeit, die parallel auch in Textform heruntergezählt wird. All dies wird über AppWizard-Interaktionen gesteuert. Die für dieses Beispiel verwendeten Werte werden aus



Bild 7. Einbinden von emWin-Code und -Ressourcen in ein bestehendes Projekt. (Bild: Segger Microcontroller)



Bild 8. Workflow mit einem AppWizard BSP (Board Support Package).

entsprechenden Variablen gelesen, die zuvor vom GUI-Entwickler im AppWizard definiert wurden. Eine Besonderheit ist dabei die Möglichkeit, bedingte Interaktionen zu spezifizieren. Der GUI-Entwickler kann festlegen, unter welchen Bedingungen – die zur Laufzeit geprüft werden – eine Interaktion ausgeführt werden soll. Einzelne Vergleiche können spezifiziert und zur Laufzeit mit einem mathematischen Ausdruck ausgewertet werden. Für Berechnungen stehen Operanden, Konstanten, Variablen oder Objektwerte zur Verfügung. Variablen sind der Schlüssel zu interaktiven GUIs. Mit Variablen können verschiedene Dinge erreicht werden: Beispielsweise können extern modifizierte Variablen als Auslöser für bestimmte Aktionen verwendet werden. Jede Variable kann mit einem individuellen mathematischen Ausdruck berechnet werden. Operanden können Konstanten, Objektwerte – Slider, Rotary usw. – oder Variablen sein. Der Ausdruck ist sehr flexibel, kann recht komplex sein und wird zur Laufzeit berechnet. Um einer Anwendung Bewegung zu verleihen, stellt AppWizard eine umfangreiche Animationsfunktion zur Verfügung. Sie kann die Eigenschaften mehrerer Elemente während der Zeitspanne einer Animation

verändern, z. B. Position, Farbe, Form, Größe usw. Auf diese Weise können komplexe Animationen erstellt werden, ohne dass Programmierkenntnisse erforderlich sind.

Filme

Die Grafikbibliothek emWin verfügt über ein spezielles Filmformat EMF (emWin Movie File Format), das drei Hauptvorteile bietet:

- Es wird nur ein Bild im Speicher benötigt
- Es werden keine Dekomprimierungs-Algorithmen benötigt
- Es fallen keine Lizenzgebühren an

Um die Konvertierung nahezu aller Filmformate nach EMF einfach zu machen, wird emWin mit einem Konverter geliefert, der auf dem Open-Source-Tool FFmpeg [1] basiert. Die kostenlose Software kann Dateien nahezu aller Filmdatei-Quellformate in jedes gewünschte Zielformat sowie in einzelne JPEG-Dateien konvertieren.

Simulation

Mit einer pixelgenauen Simulationsbibliothek auf dem PC ermöglicht emWin eine schnelle GUI-Entwicklung. Damit kann das Benutzererlebnis genau



Bild 9. Zusammenführen von User-Code und GUI-bezogenem Code. (Bild: Segger Microcontroller)

nachgebildet werden: Sowohl der Anzeigehalt, wie er auf dem Display zu sehen ist, als auch die Benutzereingaben werden für die Live-Interaktion unterstützt. Da die Simulation denselben C-Code wie das Embedded-System kompiliert und ausführt, ist das GUI-Verhalten auf dem PC dasselbe wie auf dem Zielsystem. Dies ermöglicht das Entwerfen und Debuggen der Benutzeroberfläche auf einem PC. Mit dieser Funktion können Prototypen der Anwendung zur Diskussion der Benutzeroberfläche erstellt werden.

Ein weiterer Vorteil der Simulation besteht darin, dass die GUI-Entwicklung lange vor der Fertigstellung der Hardware beginnen kann, was dazu beiträgt, die Markteinführungszeit zu verkürzen. Wenn eine Anwendung mehrere Anzeigeebenen verwendet, wie z. B. Vorder- und Hintergrund, kann emWin jede Anzeigeebene einzeln in ihrem eigenen Fenster sowie eine zusammengesetzte Ansicht anzeigen, die denselben Inhalt darstellt, und der vom Benutzer gesehen werden soll. Zur Ausführung in der Simulation wird Microsoft Visual Studio oder Code::Blocks/MinGW benötigt.

Entwickler, die mit einem Mikrocontroller eines führenden Halbleiterherstellers arbeiten, können die Grafikbibliothek emWin möglicherweise sogar kostenlos nutzen, da der Halbleiterhersteller ggf. emWin bereits für seine Mikrocontroller lizenziert hat.

Arbeiten mit dem Designtool AppWizard

Um die Grafikanwendung auf eine Zielhardware zu bekommen, stehen zwei Möglichkeiten zur Verfügung.

- Die erste Möglichkeit besteht darin, den vom AppWizard generierten Quellcode und die dazugehörigen Header-Dateien in ein bestehe-

ndes emWin-Projekt einzubinden. Der resultierende Code kann mit der gewählten IDE kompiliert und auf die Zielhardware heruntergeladen werden.

Bild 7 zeigt den Arbeitsablauf für diese Option.

- Die zweite Option (**Bild 8**) ist noch einfacher: Ebenfalls im AppWizard enthalten sind mehrere vordefinierte Board Support Packages (BSPs) für gängige Evaluierungsboards verschiedener Halbleiterhersteller. Diese BSPs ermöglichen es dem AppWizard, komplette Projekte zu generieren, meistens – aber nicht ausschließlich – für die Embedded Studio IDE von Segger Microcontroller. Dies ist der schnellste Weg, die Anwendung auf einem bestimmten Zielsystem zum Laufen zu bringen.

Die BSPs für Embedded Studio enthalten auch Binärversionen vom RTOS embOS und dem Dateisystem emFile, beide von Segger. Das vereinfacht die Auslagerung von Ressourcen auf ein externes Speichermedium erheblich. Es ist auch möglich, benutzerdefinierte BSPs innerhalb von AppWizard zu erstellen und zu verwenden.

AppWizard »Play Mode«/MSVisual-Studio-Simulation

Neben der Möglichkeit, das Verhalten der GUI auf dem PC zu simulieren, bietet AppWizard auch einen »Play Mode«, der eine schnelle Überprüfung des GUI-Verhaltens erlaubt, ohne dass Code kompiliert werden muss.

Der Unterschied zwischen dem Play Mode und dem Simulationsprojekt besteht darin, dass benutzerdefinierter Code im AppWizard Play Mode nicht Bild 9. Zusammenführen von User-Code und GUI-bezogenem Code. (Bild: Segger Microcontroller) kompiliert und daher auch nicht ausgeführt

wird. Die MS-Visual-Studio-Simulation hingegen führt auch User-Code aus.

Integration von GUI-Code und User-Code

Der einer GUI zugeordnete Code ist nur ein Teil einer vollständigen Anwendung. Um den vom AppWizard generierten GUI-Code in die vollständige Anwendung einzubinden, stellt AppWizard eine benutzerfreundliche API bereit, die vom User-Code zur Interaktion mit der GUI verwendet werden kann. Obwohl AppWizard ein sehr leistungsfähiges Tool ist, deckt es (noch) nicht alle der umfangreichen Funktionen von emWin ab. Daher kann der User-Code weiterhin reguläre emWin-API-Aufrufe verwenden, um die erforderliche GUI-Funktion zu ergänzen, die noch nicht von AppWizard abgedeckt wird. Dadurch ist es möglich, GUIs zu erstellen, die teilweise von AppWizardgeneriertem Code und teilweise von User-Code gesteuert werden (**Bild 9**).

Ausgezeichnetes GUI-Design-Tool

Mit dem ausgeklügelten GUI-Design-Tool AppWizard und der emWin-Grafikbibliothek von Segger Microcontroller ist es möglich, High-End-Benutzeroberflächen auf (Touch-)Displays zu implementieren, ohne dass solide Kenntnisse in emWin oder der Programmiersprache C erforderlich sind.

Mit der Grafikbibliothek emWin und dem Design-Tool AppWizard können Entwickler von Embedded-Systemen auf einfache Weise Smartphone-ähnliche Benutzeroberflächen erstellen. Die hier vorgestellte Backofenanwendung ist nur ein Beispiel – den Anwendungen für emWin und AppWizard sind nur durch die Phantasie des GUI-Designers begrenzt.

Das GUI-Design-Tool AppWizard von Segger Microcontroller wurde 2020 von der Design&Elektronik in der Kategorie »Development Software« mit der Auszeichnung »Innovator des Jahres« prämiert.

Axel Wolf
ist Technischer Direktor bei
Segger Microcontroller.

Literatur

- [1] FFmpeg – A complete, cross-platform solution to record, convert and stream audio and video. Fabrice Bellard, Website, www.ffmpeg.org.
- [2] Schlichtmeier, T.: With »emWin« library – User interaction like on a smartphone. elektroniknet.de, 3. Februar 2022, www.elektroniknet.de/international/user-interaction-like-on-a-smartphone.193435.html.
- [3] emWin – Graphic Library with Graphic User Interface. Segger Microcontroller, User Guide & Reference Manual, UM03001, Software Version: 6.26, Revision: 0, 4. Mai 2022, www.segger.com/downloads/emwin/UM03001.
- [4] AppWizard – Wizard for creating ready-to-use emWin applications. Segger Microcontroller, User Guide & Reference Manual, UM03003, Software Version: 1.30, Revision: 0, 4. August 2022, www.segger.com/downloads/emwin/UM03003_AppWizard.pdf.