

EMBEDDED GERÄTEENTWICKLUNG

ANFORDERUNGEN IM ZEITALTER VON SMARTPHONE,
IOT UND CLOUD

INHALT

EMBEDDED GERÄTEENTWICKLUNG

SEITE 03	Design à la Smartphone
SEITE 03	Flüssige Bedienung schafft Akzeptanz
SEITE 04	Fernbedienung mit Web und App
SEITE 04	Robuste Hard- und Softwareplattform als Gerätebasis
SEITE 05	Offen für Neues – Zukunftssichere Architektur
SEITE 06	Wolke Sieben – Geräte und IT wachsen zusammen
SEITE 07	Langlebige Geräte in dynamischen Märkten
SEITE 07	Innovation braucht starke Partner

Ob Medizintechnik, Maschinenbau oder Automobil, in jeder Branche werden heute Bedienung, Leistung und Vernetzung von Geräten mit den gewohnten Standards der allgegenwärtigen Smartphones verglichen. Maschinensteuerungen müssen cool aussehen und per Wischgesten bedienbar sein. Heizanlagen optimieren den Energieverbrauch nach aktuellen Wetterprognosen aus dem Internet und können per Smartphone bedient werden. Medizingeräte erklären sich selbständig mittels multimedialer Bedienungsanleitungen, die online aktualisiert werden. Neue Gerätefunktionen möchten Anwender bequem

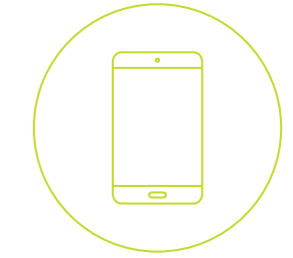
per App-Shop laden und freischalten lassen. In der Vergangenheit konnten sich die Entwickler von Gerätehard- und -software isoliert auf ihr Embedded-System konzentrieren. Heute müssen sie sich mit einer wachsenden Zahl von Themen rund um den Einsatz des Geräts beschäftigen und sich viel neues Wissen aneignen, was einen großen Aufwand bedeutet. Mit den begrenzten Ressourcen und engen Zeitplänen ist es daher sinnvoll, sich Unterstützung von einem erfahrenen Entwicklungspartner zu holen.

DESIGN À LA SMARTPHONE

Der erste Eindruck zählt und Smartphones geben zweifellos die Designtrends vor. Hochwertige schlanke Glasoberflächen sind das Maß der Dinge. Geschwungen und elegant, transparent und edel in der Erscheinung. Gerätehersteller differenzieren sich durch das äußere Erscheinungsbild. Der Einsatz im Außenbereich erfordert UV- und Temperaturbeständigkeit. Aggressive

Umgebungen in der chemischen Industrie oder auf hoher See stellen ganz besondere Herausforderungen an das Oberflächenmaterial. Eine fugenfreie, schmutzabweisende und in der Medizintechnik sogar desinfizierbare Verarbeitung ist gefordert. Nahtlose Integration des Oberflächenmaterials mit der Elektronik und Sensorik ist dafür Grundvoraussetzung. Heute werden

die elektronischen Strukturen direkt auf das Substrat gedruckt und die Elektronik wird am Glas bestückt. Zusätzliche Platinen werden überflüssig. Alle diese Anforderungen verlangen ganz spezielles Know-how über moderne Werkstoff-, Display- und Touch-Technologie. Embedded-Hardware- oder Mechanik-Entwickler bringen dieses übergreifende Fachwissen nur selten mit.



Hochwertige Bedienoberflächen erfordern ganz spezielles Know-how über moderne Werkstoff-, Display und Touch-Technologien.

FLÜSSIGE BEDIENUNG SCHAFFT AKZEPTANZ

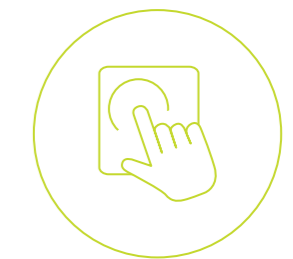
Das nächste wichtige Entscheidungskriterium für oder gegen ein Gerät ist das Bedienkonzept. Grafische Touch-Displays sind überall auf dem Vormarsch. Sie verdrängen mechanische Tasten, Schalter und herkömmliche Leuchtanzeigen, die nur noch ganz spezielle Eingabe- und Meldfunktionen übernehmen. Auch bei der Bedienung drängen die bekannten Mechanismen aus der Smartphone-Welt in die Industrie. Ansprechende Icons und intuitive, flüssige Bedienung sind die Messlatte. Der Anwender ist eine ruckelfreie Interaktion gewohnt. Wischen und Mehrfingergesten sind Standards, die auch an einer Maschine oder in einem Fahrzeug

erwartet werden. Dies steigert die Akzeptanz und reduziert den Lernaufwand beim Bediener. Die Gefahr einer Fehlbedienung wird minimiert. Farbgebung und Design der grafischen Bedienung folgen dem Design des Geräteherstellers. Auf neue Bedienkonzepte aus der Consumerwelt muss rasch reagiert werden.

All das mit selbstgestrickten Grafikbibliotheken zu realisieren ist heute nicht mehr möglich. Die Erstellung, Anpassung, Pflege und Erweiterung grafischer Bedienoberflächen in kurzen Designzyklen erfordert leistungsfähige Softwareframeworks wie Qt. Der Softwareentwickler richtet seinen

Fokus auf die Anwendung und erfindet und pflegt den dafür notwendigen Unterbau nicht immer wieder aufs Neue. Mehrsprachigkeit, Fontbibliotheken oder Stylesheets erleichtern das Zusammenstellen von Oberflächen und sichern den Markterfolg.

Da viele Embedded-Software-Entwickler sich vor allem in der hardwarenahen Programmierung mit Sprachen wie C zu Hause fühlen, erfordert die Programmierung von Oberflächen das Erlernen neuer Denkmuster und Sprachen aus der objektorientierten Programmierung. Die dafür notwendige Zeit fehlt bei der Entwicklung der eigentlichen Anwendung.



Hohe Akzeptanz und geringer Lernaufwand Wischen und Mehrfingergesten sind Standard, die auch an einer Maschine oder in einem Fahrzeug erwartet werden. Dies steigert die Akzeptanz und reduziert den Lernaufwand beim Bediener.



FERNBEDIENUNG MIT WEB UND APP

Mobile Systeme unterscheiden sich in ihrer Architektur wesentlich von Embedded-Systemen. Programmierung, Wartung und Verteilung von industrietauglichen Apps sind nicht trivial.

Web-Schnittstellen erlauben den bequemen Zugriff auf ein Gerät über das Netzwerk mittels Web-Browser. Auf diese Art und Weise können Gerätehersteller Servicepersonal oder Endkunden Zugang zu Gerätefunktionen bieten. Von der einfachen Parametrierung bis zur vollständigen Nachbildung der Bedienelemente ist alles möglich. Web-Technologien dienen heute auch zum offenen Datenaustausch mit anderen Geräten und überlagerten Softwaresystemen. Mit damit verbundenen Begriffen wie RESTful Web-APIs, JSON oder XML können klassische Embedded-Pro-

grammierer aber oft weniger anfangen als ihre Kollegen aus der IT.

Auch das Smartphone etabliert sich mehr und mehr als Bedienoberfläche für Geräte. In diesem Fall stellt der Gerätehersteller maßgeschneiderte und optisch ansprechende Apps bereit, die zumeist über Web-Mechanismen auf das Gerät zugreifen. Zur Programmierung von Apps gibt es viele komfortable Werkzeuge, mit denen man schnell vorzeigbare Ergebnisse erzielt. Für die Programmierung, Wartung und Verteilung von industrietauglichen

Apps benötigt man allerdings Expertenwissen. Auch Test und Anpassung an unterschiedliche mobile Betriebssysteme, Telefone und Tablets sind nicht trivial. Der Umgang mit den Lizenzmodellen und Programmierregeln von Apple und Co. muss verstanden werden, und die mobilen Systeme unterscheiden sich in ihrer Architektur wesentlich von den Embedded-Systemen, die der Entwickler kennt. Für viele Embedded-Entwickler ist das Neuland mit hohem Lernaufwand.



Der Einsatz moderner Chips ist ohne schlüsselfertige Softwaretools, Treiber oder Betriebssysteme wirtschaftlich nicht möglich.

ROBUSTE HARD- UND SOFTWAREPLATTFORM ALS GERÄTEBASIS

Halbleiterhersteller bieten umfangreiche Produktfamilien von Prozessoren, Mikrocontrollern, Peripheriechips, Speichern und Systems-on-Chips an, die von kleinen billigen Digitalsteuerungen bis zu leistungsfähigen 3D-Multimediasystemen die Realisierung unterschiedlicher Geräteklassen ermöglichen. Die Handbücher vieler Chips umfassen rasch mehrere tausend Seiten. Deren Flexibilität und Programmierbarkeit sind schier unbegrenzt. Aus diesem Grund

sind schon für die Inbetriebnahme der Bausteine und die Programmierung einfachster Funktionen viel Wissen und Aufwand notwendig. Bis die erste LED blinkt, können rasch Wochen vergehen. Manche Entwickler haben sich in mühsamer Kleinarbeit über lange Zeit eigene Softwarebibliotheken aufgebaut, die die Basisfunktionen der Hardware bedienen. Der Aufwand für die Wartung dieser Bibliotheken nimmt mit der Zeit zu und führt zu Ressourcenengpässen

bei der Anwendungsentwicklung. Der Einsatz moderner Chips ist daher ohne schlüsselfertige Softwaretools, Treiber oder Betriebssysteme wirtschaftlich nicht möglich. Diese Softwarekomponenten kommen von unterschiedlichen Anbietern und unterstützen oft nur einige wenige Demoplatinen. Zur maßgeschneiderten Zielhardware des Geräteherstellers sind noch viele Anpassungen und Optimierungen notwendig, um eine robuste Hard- & Softwareplattform

für die Serienproduktion zu schaffen. Die Aufwände für die Integration und laufende Wartung der Plattform über den gesamten

Produktlebenszyklus werden meistens unterschätzt. Erfahrungsgemäß binden diese wichtige Ressourcen und schränken die In-

novationskraft nachhaltig ein.

OFFEN FÜR NEUES – ZUKUNFTSSICHERE ARCHITEKTUR

Quelloffene Betriebssysteme und Softwarekomponenten wie GNU/Linux sind in der Embedded-Welt weit verbreitet. Die vielen Vorteile gegenüber hausgemachten Lösungen sind weithin akzeptiert. Alles selbst zu machen ist nicht wirtschaftlich und beeinträchtigt die Innovationsfähigkeit eines Geräteherstellers. Das wird oft viel zu spät und schmerzlich bei der Implementierung von Erweiterungswünschen oder der mühsamen Beseitigung von Fehlern erkannt. Der Fokus der Entwickler muss auf dem Kundennutzen des Geräts liegen. Die dafür notwendige Plattform und der Unterbau, wie grafische Displays, Tasteneingaben, Netzwerkanbindung, Softwareupdates, Nachladen von Funktionen oder Datensysteme, müssen nicht neu programmiert werden. Diese stehen in bewährten und getesteten Paketen zur Verfügung.

Neue Technologien und Anforderungen, wie zum Beispiel der Umstieg von IPv4 auf IPv6, können mit überschaubarem Aufwand quasi von der Stange eingebunden werden. Embedded Linux führt auf Grund seiner sauberen Struktur und Modularität rascher zum Projektziel und erlaubt es, schneller auf Änderungen zu reagieren.

Keine Software ist lückenlos fehlerfrei, auch GNU/Linux nicht. Auf Grund der Tatsache, dass tausende erfahrene Entwickler rund um den Globus ihren Beitrag zum System leisten und es sich in Unmengen an Produkten – von der Uhr bis zum Netzwerkrouter, vom Navi bis zum Roboter – im Einsatz befindet, sind die Softwarekomponenten extrem stabil. Fehler und Sicherheitslücken werden von der weltweiten Community schnell beseitigt. Diesen Grad

an Stabilität und Support würde ein Gerätehersteller für eine eigene, proprietäre Softwareplattform nur mit sehr hohem Aufwand erreichen.

Auch wenn Embedded Linux und die unzähligen Open-Source-Komponenten auf den ersten Blick kostenlos erscheinen, die Aufwände für Pflege, Wartung und laufende Aktualisierung können Projekte erheblich bremsen oder komplett zum Stillstand bringen. Um die eigene Innovationskraft zu sichern, hat es sich bewährt, die Gesamtverantwortung für die maßgeschneiderte Hard- & Softwareplattform inkl. Betriebssystem über den gesamten Lebenszyklus des Geräts in die Hände eines kompetenten Partnerunternehmens zu legen.

Die hohen Aufwände für die Integration und laufende Wartung werden meistens unterschätzt.



Fokus Kundennutzen

Der Fokus der Entwickler muss auf dem Kundennutzen des Geräts liegen.

Embedded Linux führt aufgrund seiner sauberen Struktur und Modularität rascher zum Projektziel.



Sicherer Datenaustausch zwischen Gerät und IT durch offene Softwareframeworks, unabhängig von der Kommunikationstechnik.

WOLKE SIEBEN – GERÄTE UND IT WACHSEN ZUSAMMEN

Geräte müssen heute vernetzt sein. So wie wir mit unseren Smartphones immer online sind und in den letzten Jahren neue Dienste kennen und schätzen gelernt haben, versprechen M2M, IoT, Industrie 4.0 und Cloud-Computing neue Geschäftsmodelle und Anwendernutzen in der Industrie. Die bisher separierten Welten von Geräten, Netzwerken und IT wachsen zusammen. Neuerdings müssen sich Entwickler auch Gedanken über IT-Anwendungen, Apps und Server-Hosting machen, um den Marktvorsprung zu sichern.

Am Beginn neuer Hypes stehen immer viele parallele technologische Entwicklungen, die das gleiche Ziel verfolgen. Nach längerer Zeit wird die Technologievielfalt auf einige wenige Standards bereinigt. So ist das auch im Umfeld von IoT und Industrie

4.0. Die große Menge an unterschiedlichen Netzwerktechnologien sorgt für Verwirrung. Entwickler sind überfordert und können nicht vorhersehen, welche Standards und Technologien sich letztendlich durchsetzen werden. Nichtsdestotrotz müssen sie ihr Gerät rasch auf den Markt bringen und für zukünftige politische oder technische Entwicklungen offen halten. Nur so lässt es sich erfolgreich in möglichst vielen Regionen der Welt positionieren.

In angrenzenden Gebieten wie der IT ist es für den Embedded-Entwickler noch schwieriger einzuschätzen, welche Cloud- oder Managementtechnologien zum Einsatz kommen werden und wie diese im Detail funktionieren. Umso wichtiger ist es, dass er sich auf offene Softwareframeworks verlassen kann, die den sicheren

Datenaustausch zwischen Gerät und IT unabhängig von der Kommunikationstechnik sicherstellen.

Auch Funktionen wie Fernwartung und Management ausgelieferter Geräte stellen hohe zusätzliche Anforderungen an Embedded-Entwickler. Zugriffssicherheit, Verschlüsselung und Authentifizierung müssen nachhaltig sichergestellt und laufend angepasst werden. Intuitiv bedienbare Verwaltungssoftware ist notwendig, um auch eine große Menge an ausgelieferten Geräten einfach und fehlerfrei aus der Ferne anzusprechen. Für Aufbau und Hosting von IoT- und Cloud-Plattformen benötigen Geräteentwickler IT-Partner, deren Kompetenzen über die klassische Büro- und Internetwelt hinausreichen.

LANGLEBIGE GERÄTE IN DYNAMISCHEN MÄRKTEN

Endlich geschafft, das Gerät ist fertig entwickelt und wird erfolgreich in Serie produziert. Trotz der sich rasch ändernden Ansprüche aus der Consumer-Welt verlangen industrielle Anwender eine wesentlich längere Lebensdauer ihrer Geräte. Schon beim Gerätedesign muss darauf geachtet werden, dass langlebige Bauteile, möglichst aus mehreren Quellen, eingesetzt werden. Bei einer Abkündigung muss rasch

reagiert werden, um die Integrität der Hard- und Softwareplattform sicherzustellen. Eine gute Systemarchitektur bewirkt, dass der Anwender von diesen Änderungen gar nichts merkt.

Gerätehersteller treiben einen hohen Aufwand, um Lieferketten laufend und lückenlos zu überwachen. So werden im Falle einer Abkündigung oftmals Innovationspro-

jekte abrupt gestoppt und dem Redesign des etablierten Geräts geopfert. Der Entwicklerfrust steigt und wertvoller Marktvorsprung für die Einführung neuer Produkte geht verloren. Auch hier bewährt es sich, mit einem kompetenten Partnerunternehmen über den gesamten Lebenszyklus der maßgeschneiderten Hardware-/Softwareplattform von der Entwicklung bis zur Produktion zusammenzuarbeiten.

INNOVATION BRAUCHT STARKE PARTNER

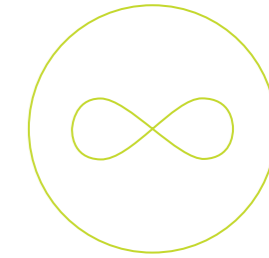
Die erwähnten Beispiele zeigen, dass Hard- und Softwareentwickler heute neben dem bekannten Zeit- und Kostendruck mit einer wachsenden Menge an Themen rund um den Einsatzbereich eines Geräts konfrontiert sind:

- Neue Trends bewerten
- Offenheit und Erweiterbarkeit sichern
- Anschluss an die IT-Welt gewährleisten
- Intuitive und flüssige Bedienung erstellen
- Neue Werkstoffkombinationen in der Elektronik nutzen
- Hochwertige Bedienoberflächen und Design entwickeln

- Unterschiedliche Programmierparadigmen und -sprachen einsetzen
- Robuste, langlebige Hard- und Softwareplattform sichern und pflegen
- Langfristig verfügbare Geräte schaffen und trotzdem rasch auf Änderungen reagieren

Zeit und Aufwand für Wissensaufbau steigen laufend. Mit begrenzten Ressourcen und engen Zeitplänen ist es unmöglich, in allen Bereichen selbst an Lösungen zu arbeiten. Innovationen im Kernbereich des Unternehmens leiden darunter. Umso wichtiger ist es, mit dem richtigen

Entwicklungspartner zusammenzuarbeiten. Als Antwort auf diese Herausforderungen bietet Ginzinger electronic systems mit seinem Partnernetzwerk Know-how und erprobte Lösungen an, damit Gerätehersteller die heutigen Herausforderungen rascher meistern und sich auf ihre Kernkompetenzen konzentrieren können.



Gute Systemarchitektur bewirkt, dass Anwender von Änderungen gar nichts merken.



GINZINGER
electronic systems

Gewerbegebiet Pirath 16 / 4952 Weng im Innkreis / T +43 77 23 54 22 / office@ginzinger.com / www.ginzinger.com