

Stefan Hoppe

# Windows 8 – die Embedded-Edition

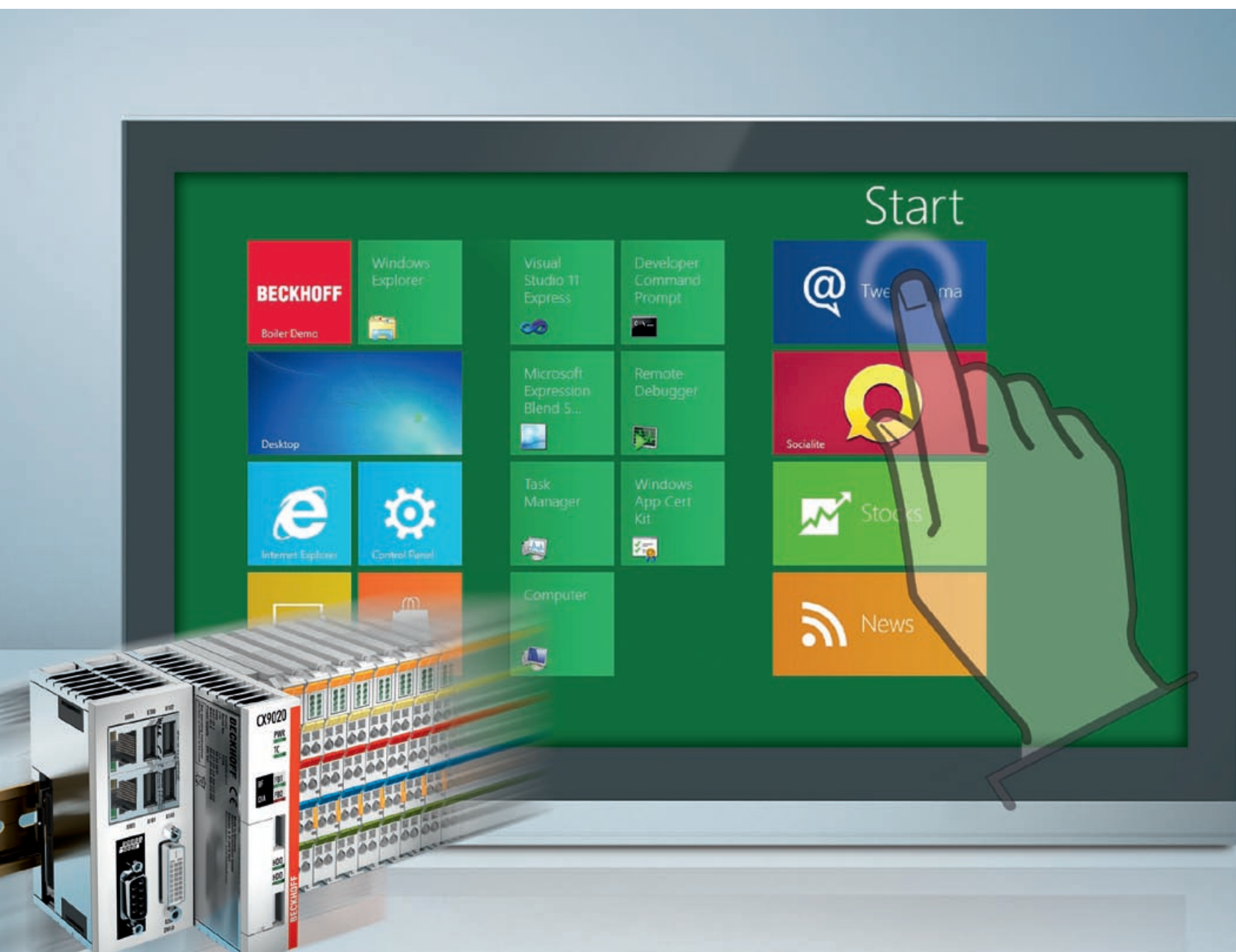
Anfang März hat Microsoft den Embedded-Ableger des Betriebssystems Windows 8 als „Community Technology Preview“ veröffentlicht: „Windows Embedded Standard 8“ oder kurz WES8-CTP. Erste Tests zeigen deutliche Verbesserungen in puncto Einbindung von Gerätetreibern, Zugriffsschutz und Branding sowie neue Möglichkeiten bei der Touch-Bedienung.

**M**icrosoft hat das neue Windows 8 im Herbst 2011 vorgestellt. Neben dem klassischen Einsatz auf PCs und Notebooks fokussiert Microsoft nun auch auf Tablets. Unabdingbar in diesem Segment ist eine andere Bedien-

philosophie als bisher: „Touch First“ lautet das Dogma. Dies wurde mit dem Metro-UI (UI: User-Interface) umgesetzt, dessen Kacheloberfläche seitdem für Aufsehen sorgt. Für den Zielmarkt Tablet-PC hat Microsoft mit einem

weiteren Dogma gebrochen: Windows 8 unterstützt nicht mehr ausschließlich die Intel x86/x64-Architekturen, sondern funktioniert darüber hinaus auf ARM-basierten System-on-Chip-Lösungen (SoC).

Generell sind die Embedded-Varianten von Windows skalierbar. Das heißt: Der Funktionsumfang des Betriebssystems lässt sich auf die Bedürfnisse der Hardware und spezielle Einsatz-Szenarien zuschneiden. Bietet WES8 auch für ARM-basierte Plattformen die Möglichkeit, Images zu erzeugen? Bisher konnte jedenfalls nur Windows Embedded Compact auf andere CPUs wie ARM skalieren. Welche Vorteile verspricht nun die neue WES8-Version für die Automatisierungstechnik, etwa hinsichtlich Durchgängigkeit der Applikationen zwischen x86- und ARM-Architekturen? Themen, die vielen



Die Kacheloberfläche von Metro-UI steht auch unter WES8 zur Verfügung. Per Shell-Launcher kann das System beliebige Shells oder Applikationen ohne Hinweis auf das im Hintergrund arbeitende Microsoft-Betriebssystem starten.

Automatisierungsanbietern unter den Nägeln brennen.

### Image-Generierung für ARM-Prozessoren?

Erste Tests im Entwicklungslabor von Beckhoff Automation zeigen, dass die öffentlich verfügbare WES8-CTP bereits einen sehr stabilen Eindruck macht. Aktuell können Images für x86/x64-Prozessoren erstellt werden; für ARM-Plattformen steht diese Funktion im CTP nicht zur Verfügung.

Windows Embedded Compact wird weiter entwickelt und nicht überflüssig, da Microsoft auch für kleinere Systeme (<1 GHz CPU-Takt und <1 GByte RAM) ein Betriebssystem anbieten will. Zudem bietet nur die Compact-Variante von Haus aus Echtzeit-Fähigkeit und ermöglicht noch kleinere Images im Bereich von 5 bis 30 MByte.

Der Umfang der WES8-Images und somit der Platzbedarf auf zum Beispiel Flash-Medien ist bei den Vorgängerversionen der Embedded-Betriebssysteme stets gestiegen und beträgt bei WES7 inzwischen 2,2 GByte bei einem allerdings sehr großzügigen Ausbau mit fast allen Komponenten: Das Image enthält dann die .Net-Frameworks 2 und 3.5, das Multi-Language-Package, den Internet-Explorer IE10, Remote-Desktop-Unterstützung (RDP), Diagnose-Tools und die Write-Filter.

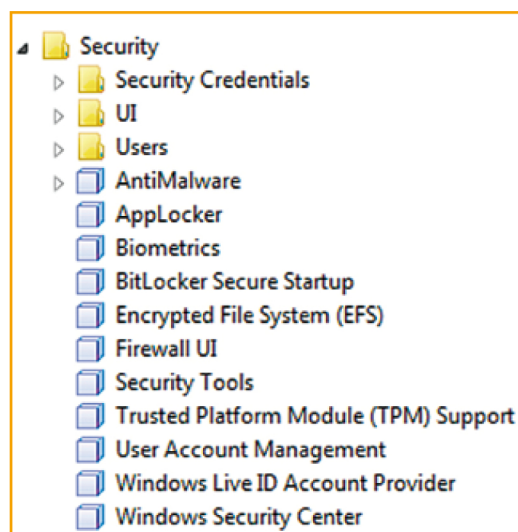
Mittlerweile spielt der Platzbedarf allerdings nur noch eine untergeordnete Rolle. Dennoch ist es erfreulich, dass WES8-CTP bei identischen Komponenten im Vergleich zu WES7 nicht mehr Speicherplatz als 2,2 GByte benötigt, trotz zusätzlicher Features wie das Metro-UI.

Die Toolkette zum Erzeugen der Images hat sich nicht verändert: Mit dem „Image Configuration Editor“ (ICE) werden wie bisher die gewünschten Funktionen des Betriebssystems festgelegt, auch Voreinstellungen wie „Write-Filter per Default nicht aktiv“. Als Er-



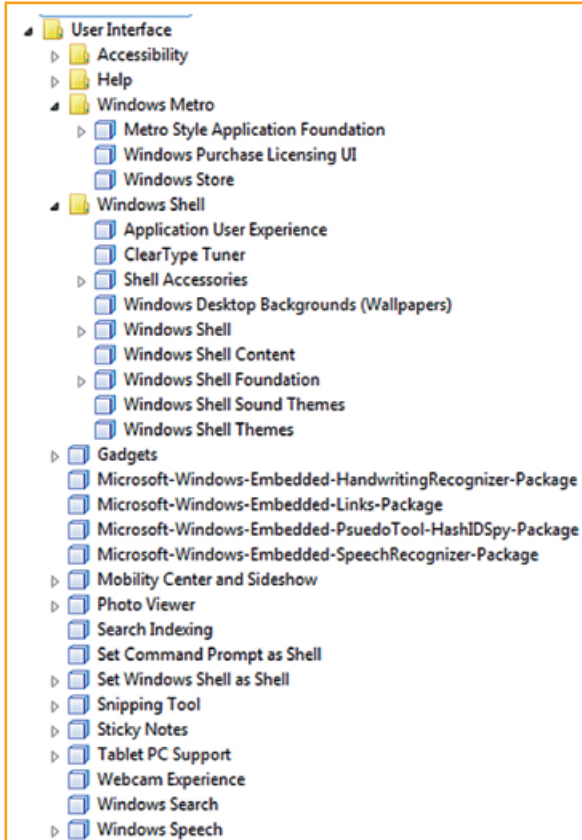
gebnis liefert der ICE ein so genanntes Answer-File im XML-Format. Wie bei WES7 wird dieses Answer-File vom „Image-Building-Wizard“ (IBW) auf einem bootfähigen USB Stick genutzt, um direkt auf der Zielhardware ein lauffähiges Image zu erzeugen.

Deutliche Verbesserungen bietet der ICE: Unter WES7 ließen sich im ICE eventuell notwendige Third-Party-Komponenten nur aus einem OEM-Verzeichnis per xcopy in das Ziel-Image integrieren. Mit WES8 und dessen neuen „Module-Designer“ erfolgt die Integration von OEM-Komponenten wesentlich einfacher: Entwickler können nun selbst Module erzeugen, die Dateien kopieren, ausführen oder auch Treiber installieren und Anpassungen an der Registry vornehmen.



Auf Hardware mit SecureBoot- und Trusted-Module-Platform (TMP) bietet WES8 deutlich verbesserte Security-Mechanismen an.

(Bilder: Beckhoff)



### WES8 im „Stealth“-Modus

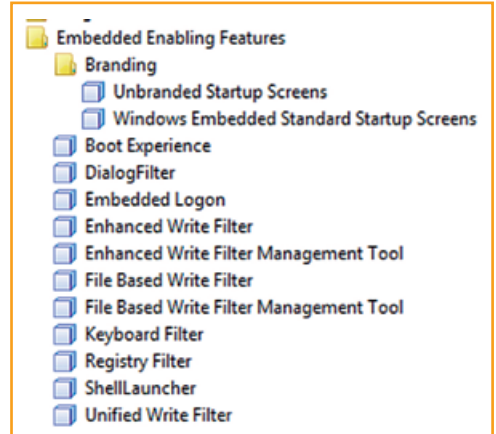
Alle Hinweise auf den Einsatz von Windows als Betriebssystem können per Branding über den gesamten Betriebszyklus eliminiert werden: Der Log-on-Bildschirm ist ebenso flexibel umzugestalten wie die Shell und der Shutdown-Vorgang. In Kombination mit UEFI-Boot (UEFI: Unified Extensible Firmware Interface) lässt sich bereits im Boot-Vorgang jeglicher Hinweis auf Microsoft komplett unterdrücken und durch eigene Grafiken ersetzen. Ein fehlender Hinweis auf „Windows“ bei Embedded-Geräten bedeutet somit nicht automatisch, dass kein Microsoft-OS im Einsatz ist.

Lag die übliche Boot-Zeit von WES7 bei 45 bis 55 Sekunden, startet WES8 aufgrund des schnelleren UEFI-Boot nun in 15 Sekunden ein Default-Image direkt in der Kacheloberfläche von Metro-UI. Von dort kann man bei Bedarf auf die bekannte Win32-Shell wechseln oder Metro-Applikationen aufrufen. Andere Startverhalten sind ebenso konfigurierbar: Der „Shell-Launcher“ erlaubt jede andere Applikation als Oberfläche zu starten. Außerdem lassen sich Oberflächen für verschiedene Benutzer oder Benutzergruppen definieren. Da-

Im „Image Configuration Editor“ (ICE) steht auch die Metro-UI-Shell als Komponente zur Verfügung. Per Shell-Launcher kann direkt in beliebige Shells oder Applikationen gestartet werden.

neben sind alle von WES7 bekannten Filter wie der Keyboard- oder der Dialogfilter zum Unterdrücken von Windows-Dialogen verfügbar. Zusätzlich gibt es den sogenannten „Gesture-Filter“, der auf Wunsch auch Touch-Aktionen abfängt.

Unter WES7 standen diverse Filter wie „Enhanced-Write“ (EWF), der „File-based-Write“ (FBWF) und der Registry-Filter zur Verfügung. Als Schreibschutz verriegelte der EWF den Speicherzugriff auf Partitionsebene. Mehr Flexibilität bot der FBWF und konnte auf Verzeichnis- und Dateiebene Ausnahmen vom Schreibschutz zulassen. Im WES8-CTP sind alle bisherigen Filter im „Unified-Write-Filter“ (UWF) mit neuen Eigenschaften zusammengefasst: Der Overlay-Speicher – ein virtueller Speicher, in dem Änderungen an Dateien/Ordern während der Laufzeit erhalten bleiben – wurde nach dem Löschen von Änderungen bisher vom EWF nicht wieder freigegeben – der UWF erledigt die Freigabe nun eigenständig. Als Registry-Exclusion des UWF steht nun der direkte Schreibzugriff in die Registry zur Verfügung, ohne ein Overlay nutzen zu müssen: Dieses Feature ist für das automatische Handling des „Day-LightSwitching“ (DST-Switching) eine enorme Erleichterung – in der Vergangenheit war es durchaus mit Aufwand verbunden, eine Lösung für das Umschalten der Sommer/Winterzeit korrekt zur erledigen: Eine eigene Applikation musste auf dem komplett verriegelten und geschützten System den Zeitsprung erkennen und vor allen anderen Applikationen beim Systemstart die Uhrzeit korrigieren. Der UWF erledigt dies und generiert für den Konfigurationszugriff ein lokales und remote erreichbares „Snap-In“ (Programm-Modul für die Verwaltung eines Computers) für die Microsoft-Management-Console (MMC) oder das scriptfähige Windows-Management-Interface (WMI).



Im Vergleich zum vollständigen Windows 8 bieten neben der Skalierbarkeit die „Embedded Enabling Features“ (EEF) die wesentlichen Embedded-Funktionalitäten. Der „Unified Write Filter“ (UWF) fasst die bisherigen Filter zusammen.

Windows 8 läuft auf allen Windows-7-Computern und stellt keine neuen Anforderungen an die Hardware. Dies gilt ebenso für die Embedded-Variante WES8-CTP, was laut erster Tests bestätigt werden kann. Die Metro-UI unterstützt die in den Automatisierungsmarkt drängenden Multitouch-Systeme sehr gut. Win32-Applikationen können wie bisher auf WES8-CTP installiert werden. Bei Metro-UI-Applikationen für die Tablet-Consumer-Welt bietet Microsoft dazu bislang nur den Weg über den konzerneigenen Marketplace an. Dies wird auch für Applikationen auf ARM-Prozessoren gelten.

Apps im Marketplace unterliegen allerdings Regeln und müssen Stabilitätsprüfungen bestehen, um eine positive „User-Experience“ in Bezug der Stabilität und Performance zu garantieren. Automatisierungsgeräte haben in der Regel aber keinen direkten Zugriff auf das Web. Hier wird es also andere Verteilmechanismen geben müssen.

WES8 als Technical-Preview macht technisch bereits zu diesem frühen Zeitpunkt einen sehr stabilen und durchdachten Eindruck. sk



**Stefan Hoppe**

ist Produktmanager Twincat bei Beckhoff Automation in Verl und „Microsoft Most Valuable Professional“ (MVP) für Windows Embedded.