

Windows 10 IoT Core auf Raspberry Pi3 Board:

Himbeere macht blau



(Bild: Shutterstock)

Microsoft setzt die Vision einer einheitlichen Windows-Plattform auch auf einem Raspberry Pi 3 Board um. Mit Windows 10 IoT Core schließt sich der Kreis – alle Windows-10-Versionen können die Services der Azure Cloud nutzen und verfügen über einen Zugang zum Internet der Dinge.

Von Marcel Bösing, Johannes Förner, Rudi Swiontek und Gerhard Biebl

Windows 10 IoT Core unterstützt zwei zentrale Konzepte: Universal Windows Driver (UWD) und die Universal Windows Platform (UWP). Die frei skalierbare Universal Windows Platform treibt die Eigenentwicklung von Anwendungen

über die gesamte Bandbreite von Windows-10-Systemen mächtig voran und besitzt in der Kombination mit dem Azure IoT Hub ein Alleinstellungsmerkmal. Mit allen Windows-10-Versionen, dazu zählen auch die Enterprise-, Mobile- und die IoT-Core Versionen, kön-

nen Geräte miteinander vernetzt werden und autonom arbeiten. Daher ist Windows 10 eine geeignete Plattform zur Entwicklung vernetzter und intelligenter IoT-Geräte (Bild 1).

Installation und Konfiguration

Die Installation und Konfiguration des aktuellen Windows 10 IoT Core auf dem Raspberry Pi 3 Board ist etwas knifflig. Aber es lohnt sich, denn die entwickelten Apps lassen sich anschließend ganz einfach von einem Desktop PC auf das Raspberry Pi 3 Board laden und debuggen. Des Weiteren gibt es eine große Auswahl an Hardware-Schnittstellen, um z.B. einen Sensor



The Embedded Experts

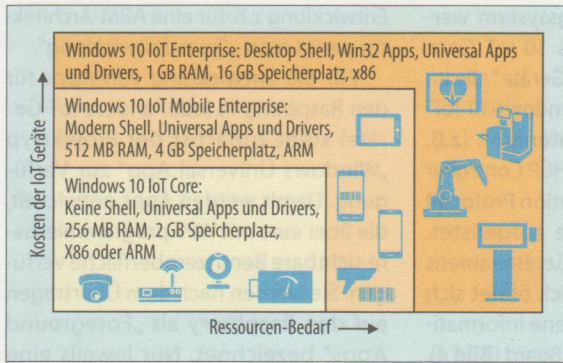


Bild 1. Windows-10-IoT-Segmentierung.

(Quelle: Avnet Silica)

anschießen zu können. Unter Verwendung des Visual Studio 2017 kann die komplette Infrastruktur für die Kommunikation mit Azure (Cloud) eingerichtet werden. Somit lassen sich Sensordaten ablegen und anschließend von unterschiedlichen Geräten aus abfragen und auswerten.

Bis jetzt war es kein Problem für Microsoft, ein Windows-System auf einen ARM-Prozessor zu portieren, da die Leistung der Embedded-Boards eher bescheiden war. Heute befindet sich auf dem Raspberry Pi 3 Model B Board eine 64-Bit Quad-Core ARMv8-CPU, die mit 1.2 GHz getaktet wird – das „non plus ultra“ für den Embedded-Bereich, vor allem in Sachen Leistung und Stromverbrauch. Allerdings stellt Microsoft nur eine 32-Bit-Variante des vorkonfigurierten Windows 10 IoT Core zur Verfügung. Jedoch wird Windows 10 IoT Core ständig weiterentwickelt, somit ist bald eine 64-Bit-Version zu erwarten. Wer darauf nicht warten möchte, kann eine eigene 64-Bit-Variante erzeugen.

Der Aufwand lohnt sich: Neuerdings gibt es für Windows 10 eine universelle Treiber-Plattform (UWP). Damit kann ein Treiber (API) für verschiedene Plattformen verwendet werden. Ein vergleichbares Angebot gibt es auch bei der Applikations-Entwicklung (Apps). Eine einmal erstellte App läuft dann auf allen Windows-10-Plattformen.

Eigenschaften der Version Windows 10 IoT Core

Die Windows-10-IoT-Core-Version ist eine abgespeckte und optimierte Windows-10-Version für kleine Geräte, die mit oder ohne User Interface (UI) betrieben wird. Es gibt kein Startmenü, keinen Launcher und nur genau eine App kann mit dem UI ausgeführt werden. Im Background hingegen können

mehrere Tasks parallel laufen. Die Steuerung kann, wie gewohnt, über Maus und Tastatur durchgeführt werden. Der Zugriff zum Raspberry ist darüber hinaus remote über eine Vielzahl unterschiedlicher Tools möglich: über den Windows 10 IOT Remote Client, das Windows 10 IOT Core Dashboard, das Webinterface welches auf dem Raspberry gehostet wird, die verschlüsselte Netzwerkverbindung SSH-Shell (Secure Shell), Power-Shell oder über das File Transfer Protocol (FTP).

Inbetriebnahme des Raspberry Pi 3

Voraussetzung für die erfolgreiche Inbetriebnahme des Raspberry Pi3 mit Windows 10 IoT Core ist ein Entwicklungsrechner mit einem Windows-10-System.

Auf den Entwicklungsrechner wird zunächst das Windows 10 IoT Core Dashboard heruntergeladen und installiert. Dieses Werkzeug dient unter anderem dazu, ein neues Gerät einzurichten. (Bild 2) Die wichtigsten Einstellungen sind dabei Geräte-Name und das Administrator-Passwort, möglichst nicht „P@ssw0rd“ denn dies ist bereits das Default-Passwort. Durch klicken auf „Herunterladen und installieren“ wird automatisch die aktuellste Version von Windows 10 IoT Core heruntergeladen und mit den gerade gewählten Einstellungen geflasht. Die verwendete SD-Karte für den Raspberry sollte unter Windows 10 IOT Core folgende Eigenschaften aufweisen: 10 MB/s Schreibgeschwindigkeit und eine Mindestgröße

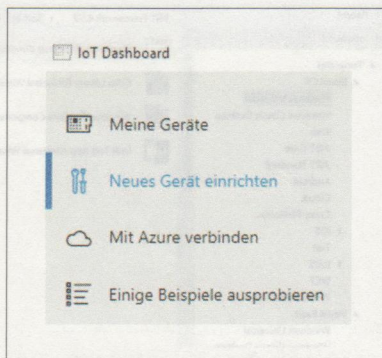


Bild 2. Das Dashboard. (Quelle: Avnet Silica)

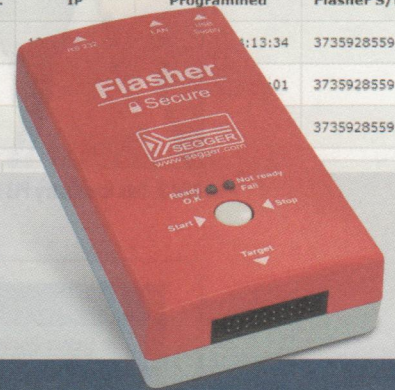


Production Programmer

Programmed devices:

No.	Serial no.	Unique device ID	Cyc
1	10000	00460019 34345109 35353835 00000000	3

No.	IP	Programmed	Flasher S/N
1		13:34	3735928559
2		01	3735928559
3			3735928559



FLASHER SECURE
Produktion mit Kopierschutz

- Komplette Infrastruktur zum Einsatz von digitalen Signaturen
- Geräte-Identifikation durch digitale Signaturen
- Transparente Produktion
- IP Protection bei Auftragsfertigung



It simply works!

www.segger.com

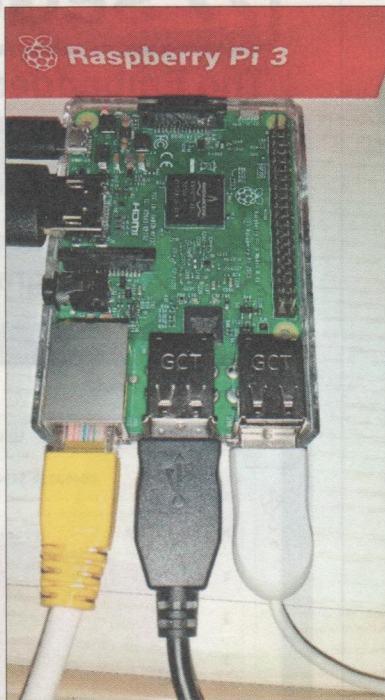


Bild 3. Das Raspberry Pi3 Model B Board.

(Quelle: Avnet Silica)



Bild 4. Der Web-Server starten mit Geräteinformationen.

(Quelle: Avnet Silica)

von 8 GB. Wer auf Nummer Sicher gehen möchte, kann eine der Karten nutzen, die von Microsoft zertifiziert und empfohlen werden: Dies ist einmal die Samsung EVO 32GB Class 10 Micro SDHC Card und von SanDisk die Ultra Micro SDHC 16GB Card.

Das Raspberry Pi 3 Board wird ähnlich wie ein PC aufgebaut und angeschlossen (Bild 3). Anschließend wird das Raspberry Board mit der hinzugefügten SD-Karte gebootet. Der erste Boot-Vorgang kann einige Zeit in Anspruch nehmen. Der anschließend erscheinende Willkommensbildschirm, die sogenannte Startup-App, zeigt die vorher festgelegten Einstellungen, wie den Gerätenamen und Netzwerkinfos an. In der oberen Menüleiste kann über den Reiter „Device Settings“ die Sprache oder das Netzwerk konfiguriert werden.

Auf dem Entwicklungssystem werden mit dem Windows 10 IoT Core Dashboard unter „Meine Geräte“ alle im Netzwerk befindlichen Windows-10-IoT-Core-Systeme mit Gerätenamen (z.B. rs_rasp3) und der vom DHCP Controller (Dynamic Host Configuration Protocol) vergebenen IP-Adresse aufgelistet. Durch Auswählen eines Gerätenamens und mit einem Rechtsklick öffnet sich ein Dialog mit verschiedene Informationen über das Raspberry Board (Bild 4). Über den Menüpunkt „Im Device Portal öffnen“ wird der integrierte Webserver gestartet. Hier findet man verschiedene Informationen über das Windows-10-IoT-Core-System wie den Performance- und Ressourcen-Bedarf. Außerdem wird hier vermerkt, welche Prozesse gerade laufen und welche Treiber sowie andere Hardware-Komponenten installiert sind. Zusätzlich lassen sich an dieser Stelle Apps installieren und administrative Tasks ausführen, vergleichbar mit dem Systemsteuerungsmenü unter Windows 10.

Auch ohne externe Zusatzgeräte wie Monitor, Maus und Tastatur kann der Raspberry konfiguriert und bedient werden. Microsoft stellt dafür im App Store ein Tool namens Windows IOT Remote Client zum kostenlosen Download bereit, mit dem alle Funktionen des Raspberry remote über den Entwicklungsrechner benutzt werden können.

Installation von Visual Studio 2017 Community

Die Community-Version von Visual Studio 2017 ist kostenlos und ermöglicht es dem Entwickler, IOT -Anwendungen in Form von „Apps“ für das Raspberry Pi 3 Board zu entwickeln. Solche Apps können in C#, C++ oder Visual Basic verfasst werden. Diese Art der Software-

Entwicklung z.B. für eine ARM-Architektur nennt man „Cross-Entwicklung“.

Für die Entwicklung von Apps für den Raspberry Pi (oder andere IoT-Geräte) steht zunächst der Projekttyp „Windows Universal App“ zur Verfügung. Damit werden Apps entwickelt, die über eine mit WPF programmierbare sichtbare Benutzeroberfläche verfügen. Sie werden nach dem Übertragen auf den Raspberry als „Foreground Apps“ bezeichnet. Nur jeweils eine dieser Foreground Apps hat exklusiven Zugriff auf den „Bildschirm“, daher kann auch immer nur eine solche App ausgeführt werden. Die App funktioniert auch, wenn kein realer Bildschirm an den Raspberry angeschlossen ist; die Benutzeroberfläche kann dann über den Remote Client auf dem Entwicklungsrechner sichtbar gemacht werden.

Alternativ dazu können auch sogenannte „Background Apps“ entwickelt werden. Dies setzt aber voraus, dass die dafür benötigten Visual-Studio-Projekt-Templates installiert werden. Diese werden als Download unter „Windows IoT Core Project Templates“ zur Verfügung gestellt.

Foreground und Background Projekte erstellen

Um eigene Apps zu entwickeln muss das Entwicklungssystem in den sogenannten „Entwicklermodus“ umgestellt werden. Unter Einstellungen → Für Entwickler lässt sich der Modus aktivieren.

Unter File → New → Project ... wird ein Visual-C#-Windows-Universal-Projekt mit dem Template „Blank App (Universal Windows)“ ausgewählt (Bild 5). In diesem leeren Rahmenprojekt unter TODO kann der eigene Programmcode hinzugefügt werden. Das Testen und Debuggen erfolgt remote

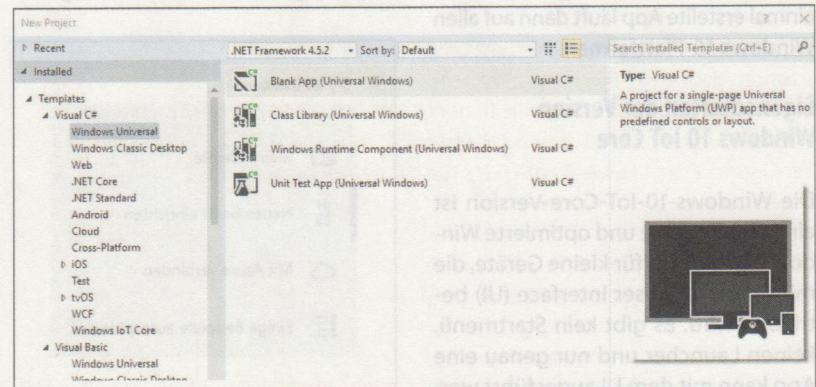


Bild 5. Mit Hilfe des Templates lassen sich neue Foreground Apps entwickeln.

(Quelle: Avnet Silica)

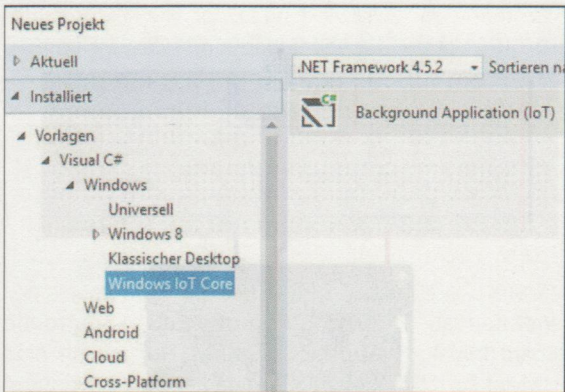


Bild 6. Neues Background-Projekt anlegen.

(Quelle: Avnet Silica)

direkt am +3,3 V- oder +5 V-Kreis funktionieren kann. Aber das sollte nicht die Regel sein, denn von einem dauerhaft sicheren Betrieb kann man nicht ausgehen. Die Stromentnahme an einem GPIO-Ausgang muss in jedem Fall begrenzt werden. Wenn ein GPIO-Ausgang etwas schalten soll, ver-

über die Ethernet-Schnittstelle. Dazu muss der Remotecomputer anhand des Gerätenamens oder der IP-Adresse ausgewählt werden. Danach kann das Projekt auf dem Remotecomputer bereitgestellt und ausgeführt werden. Durch Starten des Debuggers wird das Projekt erzeugt und zum Testen auf das Gerät übertragen. Zum Debuggen kann an einer beliebigen Stelle im Programmcode der Cursor platziert werden und mit der Einstellung „Ausführen bis Cursor“ wird die Anwendung bis zur Markierung ausgeführt. Eine andere Möglichkeit ist das Setzen von Breakpoints im Programmcode.

Eine Background Application für Windows IOT Core wird erstellt unter: File → New → Projects → Windows IOT Core → Background Application (Bild 6).

Ansteuerung externer Hardware durch den Raspberry Pi

Neben den vier USB-Schnittstellen und der HDMI-Schnittstelle bietet der Raspberry Pi 3 die Möglichkeit, externe Geräte wie zum Beispiel Sensoren, LC-Displays und LEDs über eine 40-polige Steckerleiste anzusteuern.

Die Stecker stellen verschiedene Spannungen und Anwendungsmöglichkeiten bereit. An Pin 1 und 17 liegt eine Spannung von +3,3 V an. Pin 2 und 4 führen dagegen +5 V. Insgesamt acht Ground-Anschlüsse stehen mit den Pins 6, 9, 14, 20, 25, 30, 34 und 39 zur Verfügung. Damit verbleiben 28 Pins für die externe Kommunikation (Bild 7).

Die Pins mit fest definierter Spannung eignen sich nicht zur Stromversorgung. Externe Schaltungsteile, insbesondere Relais und Motoren, sollten über ein eigenes Netzteil versorgt werden. In Ausnahmefällen mag es möglich sein, dass ein Relais oder auch ein Lüfter

gestärkt man das Ausgangssignal mit einer Transistorstufe oder einer Treiberschaltung, um den GPIO zu entlasten. Der Gesamtstrom aller GPIOs sollte 50 mA nicht übersteigen. Die gesamte Elektronik eines einzelnen Anschlusses ist für maximal 16 mA ausgelegt, bis dahin treten keine Schäden auf. Alles darüber ist unsicher. Eine gute Regel ist, die Ausgangsendstufen eines GPIO zwischen 2 mA und 16 mA zu halten.

Neben den sog. GPIO Pins zur parallelen Kommunikation stehen insgesamt drei unterschiedliche serielle Kommunikationsprotokolle zur Verfügung:

- SPI (Pin 9, 10 und 11)
- I²C (Pin 27 und 28)
- UART (Pin 8 und 10)

Sowohl GPIO als auch SPI, I²C und UART werden durch eine Anzahl von Klassen in entsprechenden Namespaces (Windows.Devices) unterstützt. Hierzu muss allerdings zuerst mit Nuget die Bibliothek „Windows.Devices“ in Visual Studio installiert werden. Dadurch werden auch eine Reihe weiterer interessanter Namespaces eingefügt, wie z.B. Bluetooth, Sensors und WIFI.

Raspberry Pi 3 Model B/B+			
3V3	1	2	5V
GPIO2	3	4	5V
SDA1 I ² C	5	6	Ground
GPIO3	7	8	GPIO14
SCL1 I ² C	9	10	UART0_TXD
GPIO4	11	12	GPIO15
Ground	13	14	UART0_TXD
GPIO26	15	16	GPIO20
Ground	17	18	GPIO21

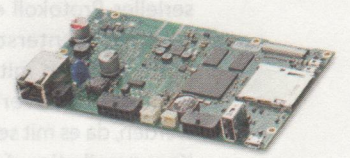
Bild 7. Die 40 I/O-Pins vom Raspberry Pi 3.

(Quelle: Avnet Silica)



FITS IN ANY ENVIRONMENT
by Garz & Fricke

Reliable Quality Made in Germany



Single Board Computer and Human Machine Interface

- Complete system with CPU board, display, touch, front glass and housing
- Freescale ARM®i.Mx6 architecture
- Scalable CPU performance
- Ready-to-run systems
- Industrial solutions
- Board support packages with drivers for all interfaces
- Operating systems: Windows Embedded Compact®, Linux and Android™

SOLUTIONS THAT COMPLETE!

GARZ & FRICKE

Garz & Fricke GmbH | Hamburg | Germany
info@garz-fricke.com | www.garz-fricke.com

Anwendung in der Praxis

Eine Anwendungsmöglichkeit des Raspberry Boards mit Windows 10 IoT Core ist die Nutzung als Messapparat für das Raumklima. Dazu werden Temperatur- und Luftfeuchtigkeitsmesswerte von einem Sensor eingelesen, auf dem User Interface angezeigt, auf der SD-Karte zwischengespeichert und letztlich in Microsoft Azure als Messages an ein Azure IOT Hub weitergeleitet. Von dort können die Telemetriedaten von beliebigen Endgeräten abgefragt werden.

Der DHT11-Sensor ist in zwei Varianten, mit drei oder vier Anschlüssen, verfügbar. Bei der dreipoligen Bauform, bei welcher der Sensor auf einer kleinen Platine untergebracht ist, ist meist ein Vorwiderstand mit verbaut. Bei der alternativen Bauform muss dieser extern beschaltet werden (Bild 8). Der nicht beschaltete vierte Anschluss sollte auf Gnd gelegt werden.

Die meisten Sensoren verwenden, soweit diese nicht einfach binär über GPIO angesteuert werden können, das SPI- oder I²C-Protokoll. Es gibt aber auch Sensoren, wie den weit verbreiteten kombinierten Temperatur- und Luftfeuchte-Sensor DHT11 bzw. DHT22, die ein abweichendes proprietäres serielles Protokoll einsetzen, das von .Net nicht unterstützt wird. Dieses Protokoll kann mit C# nicht in voller Bandbreite zuverlässig umgesetzt werden, da es mit sehr engen zeitlichen Kommunikationsfenstern arbeitet – zwischen zwei und 80 Microsekunden. Diese Werte können nicht immer zuverlässig eingehalten werden, da C# nicht deterministisch arbeitet. In diesem Fall wählt man den Weg über eine in C++ geschriebenen Bibliothek, die das Protokoll implementiert.

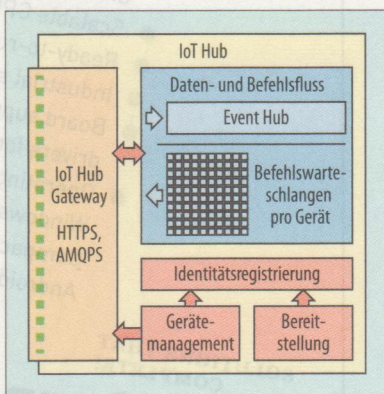


Bild 9. Schematische Darstellung des Azure IoT Hubs. (Quelle: Avnet Silica)

Ein einfaches in WPF verfasstes User Interface übernimmt die Anzeige der gemessenen Daten. Diese können zeitgleich auf einem über I²C angesteuerten zweizeiligen LC-Display sichtbar gemacht werden.

Die SD-Karte von der das Betriebssystem des Raspberry gebootet wurde, kann auch als Speichermedium für Messdaten benutzt werden. In Fällen, in denen ein Internetanschluss nur temporär verfügbar ist, kann es sehr sinnvoll sein, die Daten zu puffern. Das Speichern von Daten auf der SD-Karte ist allerdings nicht direkt vergleichbar mit dem Arbeiten mit Dateien in einer „normalen“ .NET-Desktop-Anwendung, da der Zugriff auf den Datenträger nur eingeschränkt über eine virtuelle „Sandbox“ möglich ist. Aus diesem Grund kann man auch nicht die üblichen Klassen des Namespace System.IO einsetzen.

Das eigentliche Ziel ist aber, die gemessenen Telemetrie-Daten möglichst unmittelbar über eine Internetverbindung an den Azure IOT Hub in Form von sogenannten „Messages“ weiterzuleiten. Dann können weltweit von allen möglichen Endgeräten aus auf diese Messages zugegriffen werden.

Microsoft Azure

Microsoft Azure ist aus IT-Sicht ein fast schon omnipotentes Konglomerat aus verschiedenen Dienstleistungen, die bei Bedarf gebucht werden können und weltweit zur Verfügung stehen. Von der Bereitstellung von IT-Infrastruktur über Datenbank-, Website- und Webservice-Hosting bis hin zu einfachen Telemetrie-Messaging-Diensten wie Azure IOT Hub wird alles angeboten. Microsoft arbeitet dabei mit dem „Pay as you go“-Modell, bei dem nur in Anspruch genommene Leistungen bezahlt werden.

Azure IoT ist Microsofts Public-Cloud-Plattform für Entwickler und Administratoren, die eigene Software-

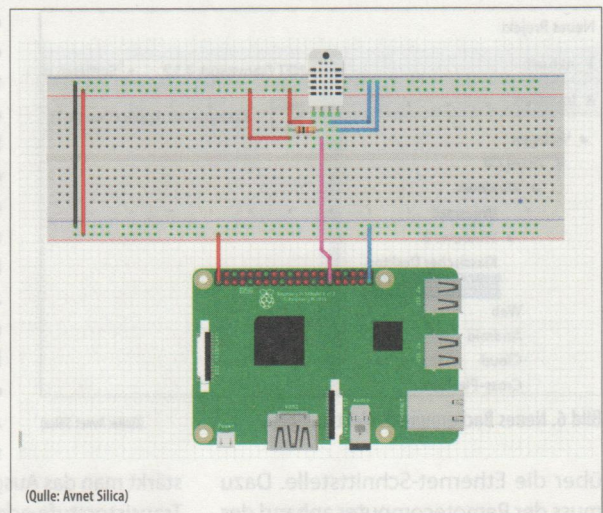


Bild 8. Beim vierpoligen Sensor muss ein Widerstand mitverbaut werden. (Quelle: Avnet Silica)

Anwendungen in der Cloud ausführen möchten. Die Plattform umfasst eine Reihe von Diensten, die einzeln oder in Kombination genutzt werden können. Azure IoT Hub bietet dabei eine einfache und sichere Möglichkeit, Millionen von IoT-Geräte anzuschließen, bereitzustellen, zu verwalten und mit diesen bidirektional zu kommunizieren (Bild 9). Der IoT Hub ist die Brücke zwischen Geräten und entwickelten Lösungen in der Cloud, sodass Daten in Echtzeit gespeichert und analysiert werden können. Dabei werden eine beliebig große Speicherkapazität sowie andere Dienstleistungen bereitgestellt und nutzungsabhängig abgerechnet (automatische Skalierung).

Beim IoT Hub erfolgt die Authentifizierung auf Gerätebasis, damit in jeder IoT-Lösung die gewohnte Vertraulichkeit sowohl von Cloud-zu-Gerät- als auch von Gerät-zu-Cloud-Nachrichten beibehalten wird. Des Weiteren unterstützt die Microsoft Azure Cloud sichere Kommunikationsstandards. TLS (Transport Layer Security) wird immer vorausgesetzt, sodass man sich über HTTPS (HyperText Transfer Protocol Secure) und AMQPS (Advanced Message Queuing Protocol) direkt mit dem Azure IoT Hub verbinden kann. Ist dies nicht möglich, hat man die Möglichkeit, seine Kommunikation mit Hilfe eines Gateways in die entsprechenden Protokolle zu übersetzen.

Alle Geräte können mithilfe der Open-Source-Geräte-SDKs mit Azure IoT verbunden werden. Dabei unterstützen die SDKs verschiedene Betriebssysteme wie Linux, RTOS, Windows und Echtzeitbetriebssysteme sowie mehre-

Hardware Raspberry Pi 3 B

Der Raspberry Pi in der Version 3 B verfügt über eine vergleichsweise leistungsfähige Hardware. Durch entsprechende Erweiterungen (Datenträger, Monitor, Tastatur) kann er zu einem vollwertigen Mini-Computer ausgebaut werden. Seine technischen Daten im Einzelnen:

- Broadcom BCM2837 1200 MHz 64 Bit ARM-Cortex-A53 64-Bit-Quad-Core-Prozessor mit VideoCore IV Dual-Core-GPU
- GPU bietet Open GL ES 2.0, OpenVG mit Hardware-Beschleunigung und 1080p30 H.264 High Profile Decoding
- GPU ermöglicht 1 GPixel/s, 1,5 GTexel/s oder 24 GFLOPs mit Textur-

- filterung und DMA-Infrastruktur
- 1 GB RAM, extern erweiterbar mit Flash (8/16/32/64 GB) / RAM / ROM
- 4-polige Audio/Composite-Video-OUT-Buchse, 3,5 mm
- 15-poliger MIPI-CSI-2-Steckverbinder für Raspberry-Pi-HD-Videokamera
- 15-poliger serieller Display-Schnittstellensteckverbinder
- HDMI-Buchse für externen Monitor
- 4 USB-Anschlüsse
- Stromversorgung: 5V, ca. 2,5 A max. über MicroUSB-Buchse
- 10/100 Ethernet-LAN-Anschluss
- WLAN und Bluetooth onboard
- 40 IO Ports über GPIO-Pfostenstecker, davon einige nutzbar für serielle Kommunikation: SPI, I²C, UART

re Programmiersprachen wie C++, Node.js, Java, C#, .NET und Python.

Voraussetzung für die Nutzung der Microsoft-Azure-Service-Plattform ist eine Registrierung und die Hinterlegung einer Kreditkarte.

Der starke Partner für Azure IoT

Avnet Silica verfügt bereits über ein Konto und stellt sich damit als Azure Cloud Solution Provider (CSP) für industrielle Zwecke zur Verfügung. Mit der Avnet-Lösung erhalten Kunden den kompletten Zugriff auf die Microsoft-Azure-Services und bekommen eine detaillierte monatliche Abrechnung.

Ebenfalls stellt sich Avnet Silica als Partner für Lösungen im Bereich des Azure IoT Hubs oder der IoT Suite bereit und, um die einzelnen Geräte bei Azure anzumelden. Natürlich sind weiterhin die Microsoft-Embedded / IoT-Lizenzen sowie Sever-Lizenzen für den industriellen Gebrauch über Avnet Silica beziehbare.

Die Sicherheit der IoT-Lösungen ist durch eine Authentifizierung auf Geräteebene gewährleistet, sodass nur unter Verwendung der richtigen Anmeldeinformationen mit diesen Geräten kommuniziert werden kann. So lassen sich die Möglichkeiten des IoT voll ausschöpfen.

Die im Beitrag dargestellte Beispielanwendung kann über www.hackster.com im Quellcode heruntergeladen werden.



Marcel Bösing

ist Technical Manager beim Microsoft Embedded Distributor Avnet Silica und dort unter anderem für Design-In-Unterstützung und Beratung zuständig. Avnet Silica vertreibt die genannten Windows-10-IoT-Lizenzen.

marcel.boesing@avnet.eu



Johannes Förner

leitet die Bereiche Marketing und Vertrieb bei der Private ptm-Akademie in München. Er verantwortet auch die Entwicklung neuer Schulungsprodukte, wie den Kurs „Windows

10 IoT Core für Raspberry Pi 3 und Anbindung an Azure IoT Hub“, der in Zusammenarbeit mit Rudi Swiontek und Gerhard Biebl entstand.

johannes.foerner@ptm.de

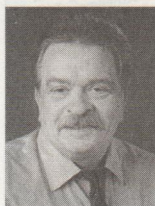


Rudi Swiontek

ist im Bereich Schulung und Entwicklung tätig. Als erfahrener Produktspezialist ist er auch für die Windows-Embedded-Betriebssysteme zuständig und bietet u.a. Windows-

10-IoT-Trainings an.

rs@it-swiontek.de



Gerhard Biebl

ist seit mehr als 37 Jahren als Software-Entwickler und Dozent freiberuflich tätig und ist spezialisiert auf die OOP-Programmiersprachen C++ und C#, für die er eine Reihe von

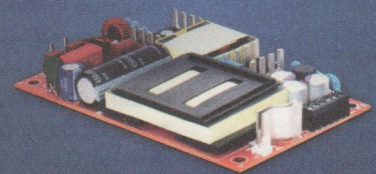
Schulungen anbietet.

bieblsoft@aol.com



THE NEW (M)ULP ULTRA LOW PROFILE SERIES

THE WORLD'S SMALLEST AND FLATTEST POWER SUPPLY



(M)ULP275

- 5 x 3 x 0.75 INCHES
- 150 W CONVECTION-COOLED
- EFFICIENCIES UP TO 93%
- BF RATING FOR MEDICAL VERSIONS
- PG/PF SIGNAL OPTION
- 12 V/0.5 A FAN OUTPUT STANDARD

