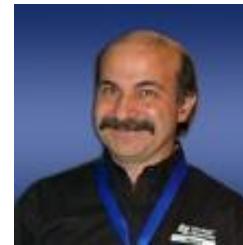


TTG – INCONTRO DEL 27 GIUGNO 219

AZURE SPHERE:
Facile da usare,
Disegnata per essere sicura!

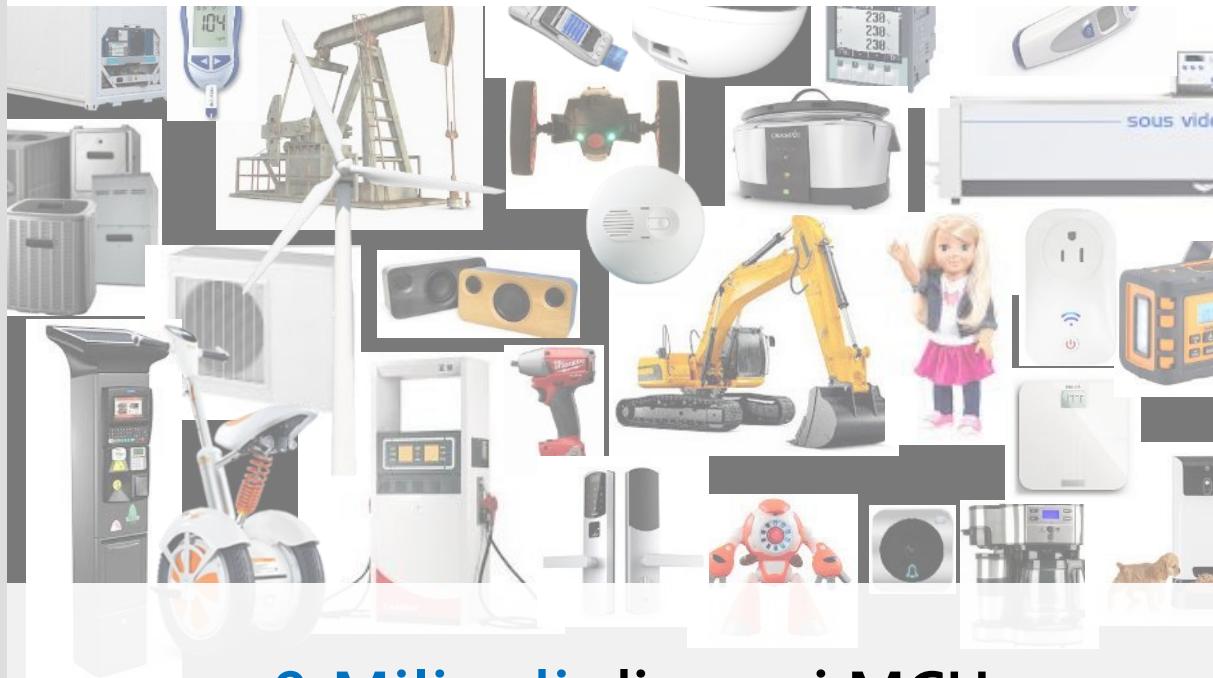
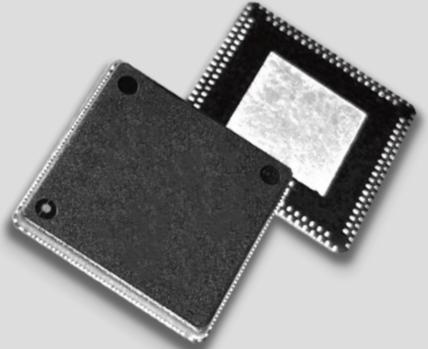


Beppe Platania

CEO di BEPS Engineering S.r.l.

Twitter: @beppeplata
Blog: beppeplatania.com
eMail: beppe.platania@bepseng.it

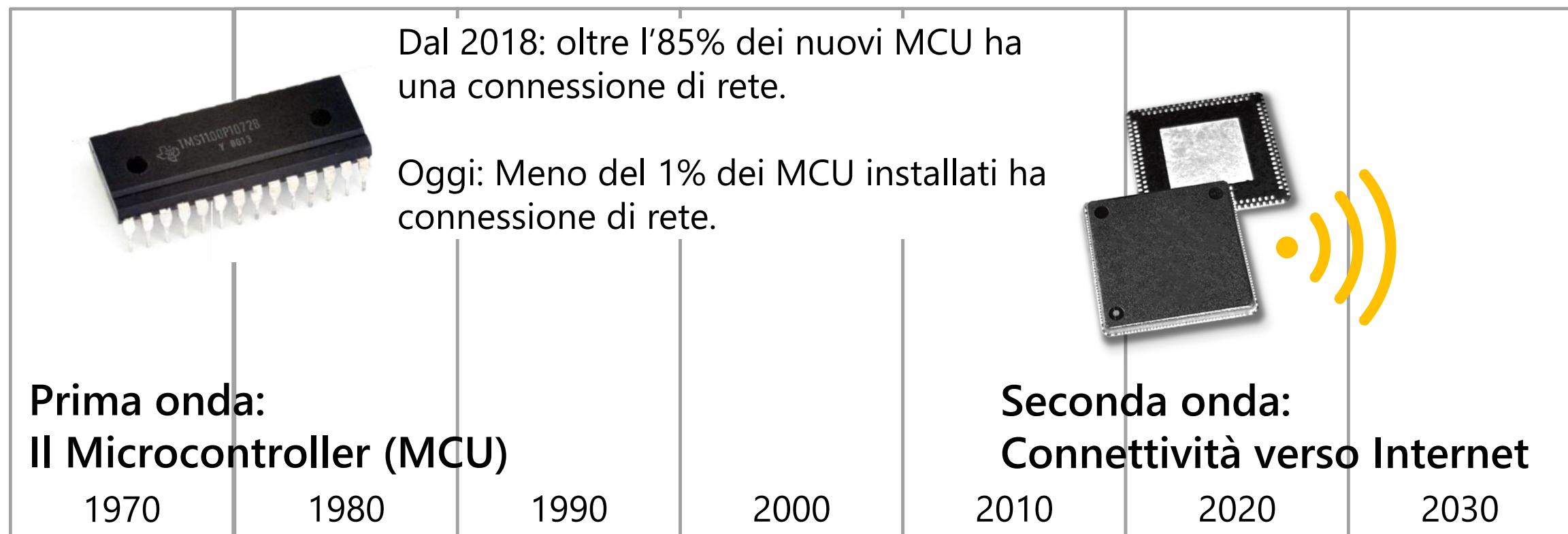
Microcontrollori (MCUs) low-cost, single chip computers



9 Miliardi di nuovi MCU
immessi sul mercato ogni anno



Prepariamoci per la seconda onda della trasformazione digitale



Qual è il vantaggio di avere MCU «connesso» in un frigo



Come fa un utente ad accorgersi che il compressore del frigo va sostituito?

Opzione 1

Surgelati sciolti, latte avariato
Richiesta al supporto per la sostituzione del compressore

Opzione 2

Messaggio dal supporto tecnico per la sostituzione del compressore (in giornata)

Il frigo così connesso ci offre una migliore esperienza.

... lo si faceva anche prima di Internet! Cosa Cambia?

Internet

Opportunità

Rischi



Opportunità

**Facilità di collegamento
Velocità (Fibra, 4G, 5G)
CLOUD
Moderne Infrastrutture
Intelligenza Artificiale
...**

Rischi

**Identità del dispositivo
Identità della sala server
Intrusioni sul collegamento
...**

Internet non è una “seriale” molto lunga !

"Ransomware attacks will target more IoT devices in 2018"

"When smart gadgets spy on you: Your home life is less private than you think"

"Hackers infect 500,000 consumer routers all over the world with malware"

"Hacking critical infrastructure via a vending machine? The IOT reality"

"Huge IoT botnet may be used for Ukraine attack"

"Industrial IoT to equip new era of corporate intruders coming in through devices"

"Security experts warn of dangers of connected home devices"

"Hacking these IoT baby monitors is child's play, researchers reveal"

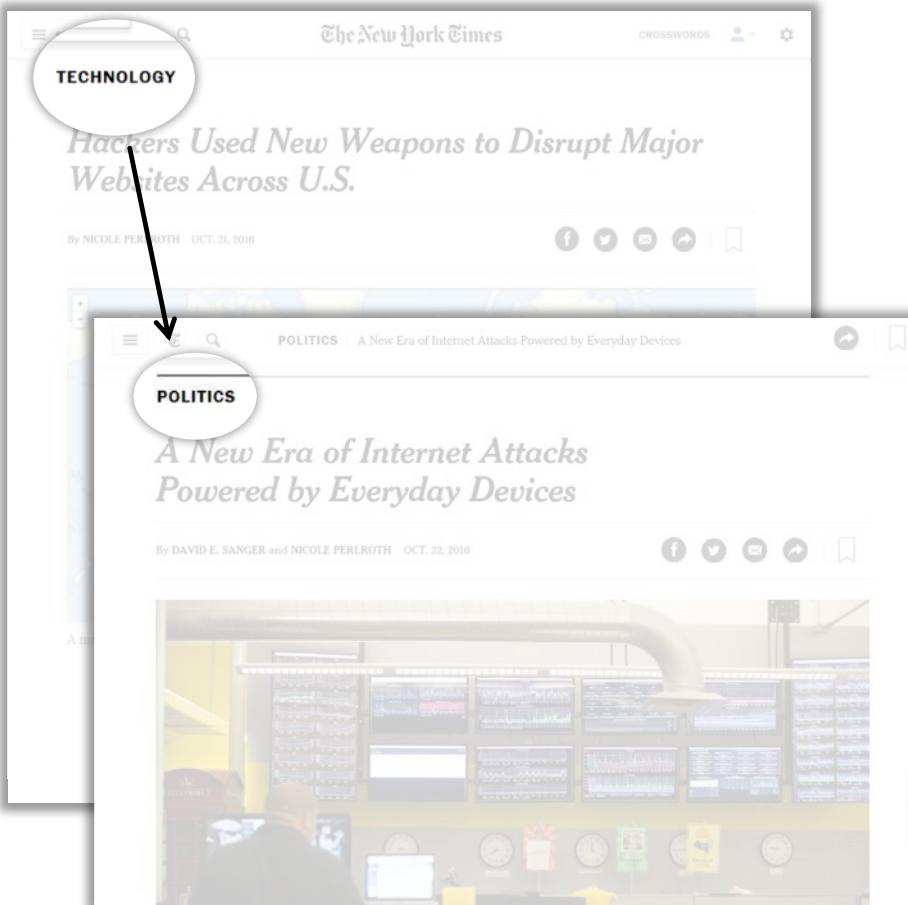
"Your smart fridge may kill you: The dark side of IoT"

"The Lurking Danger of Medical Device Hackers"

"Protecting Your Family: The Internet of Things Gives Hackers Creepy New Options"

"Why the KRACK Wi-Fi mess will take decades to clean up"

IoT e sicurezza: 21 Ottobre 2016 "Mirai" Botnet Attack



Osservazioni



La sicurezza IoT è un problema socio-economico
Primo giorno l'attacco è nella pagina **Tecnologica** del New York Times
Secondo giorno l'attacco passa nella pagina **Politica**



Il pericolo è reale, specialmente per i dispositivi IoT



Mirai Botnet (Dyn Attack), Ott 2016: Il più vasto attacco IoT/DDoS (Distributed denial of service attack). Una gran parte di Internet indisponibile: Twitter, the Guardian, Netflix, Reddit e CNN. Dispositivi colpiti: Webcam e DVR.

Jeep Hack, Luglio 2015: Un team di ricercatori è riuscito a prendere il totale controllo di una Jeep sfruttando le vulnerabilità di aggiornamento del firmware



TRENDnet Webcam Hack, Gen. 2012. Acquisizione all'accesso alla videocamera e al microfono via TCP/IP.



Hackable Cardiac Device, Gen. 2017: la vulnerabilità ha fornito un accesso per consumare le batterie, cambiare il battito e creare shock!



Printer Hack, Nov. 2011: Surriscaldando il fusore, veniva causata l'accensione della carta della stampante.

Torniamo al nostro frigo connesso ad Internet



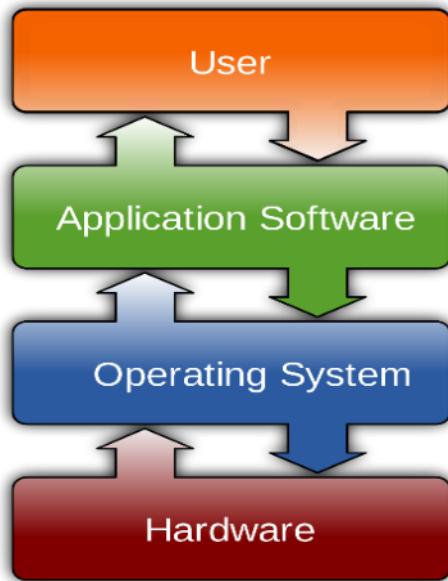
From: Hackers
To: Consumer
Subject: Your Fridge

We control your fridge.
Send us \$5 in bitcoin or else...

... Nessun produttore vuole fornire dispositivi a rischio.

. . . Peggio: pensiamo ad una stufa a GAS

- In Germania (2016) si contano quasi 8 milioni di stufe a GAS
- In Irlanda (2016): il 34% delle case si riscalda con stufe a GAS.



La differenza tra la stufa e una bomba è legata soltanto al software!
Apri la valvola del gas, aspetta 5 minuti, dai **fuoco** !!

La battaglia sulla sicurezza di internet

Microsoft ha combattuto in questo campo da decenni ed ha tante esperienze da condividere.
Anche a livello hardware!

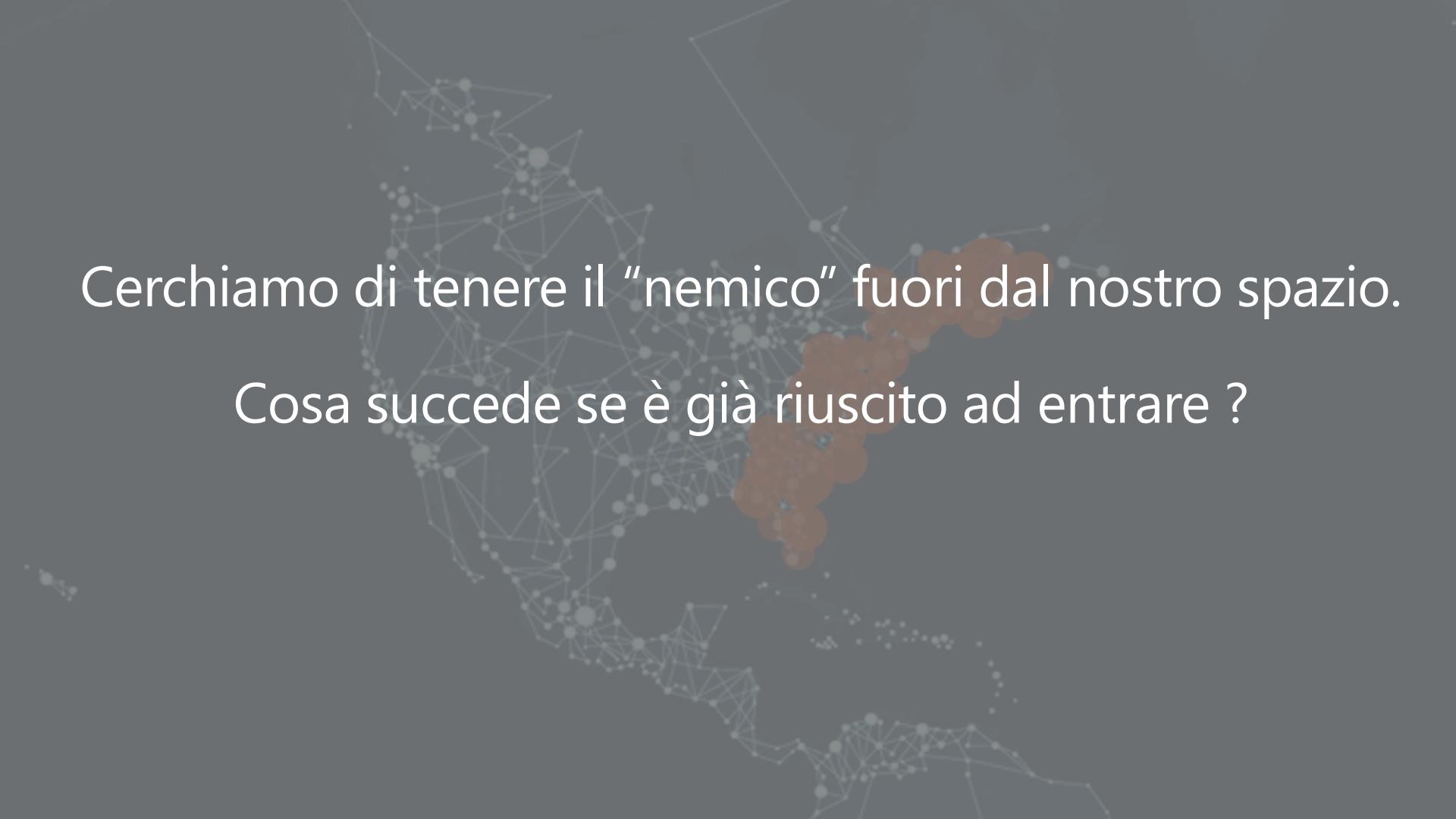
Portiamo ad esempio l'XBox:

XBox: "Hackerato" in settimane
-> Sistema Intel x86 standard

XBox 360: "Hackerato" in pochi mesi
-> Intervento HW per "Hackerare" il BUS

XBox One: Fino ad oggi non è stato "Hackerato"
-> anche grazie all'inserimento di un firewall interno al chip





Cerchiamo di tenere il “nemico” fuori dal nostro spazio.

Cosa succede se è già riuscito ad entrare ?



La sicurezza è fondamentale

Per questo va costruita dall'inizio !



Torino
Technologies
Group

Le sette proprietà per un dispositivo altamente sicuro



Hardware Root of Trust



L'identità del dispositivo e l'integrità del software sono garantite a livello hardware?



Defense in Depth



Il tuo dispositivo rimane protetto se un meccanismo di sicurezza viene neutralizzato?



Small Trusted Computing Base



La comunicazione del tuo dispositivo è protetta da bug in altri codici?



Dynamic Compartments



La sicurezza del tuo dispositivo è organizzata a compartimenti stagni?



Certificate-Based Authentication



Per l'autenticazione si usano certificati invece di password?



Failure Reporting



Il tuo dispositivo segnala guasti e/o anomalie?



Renewable Security



Il tuo dispositivo si aggiorna automaticamente?

Alcune proprietà
dipendono soltanto
dall'hardware



Hardware
Root of Trust

Hardware Root of Trust

Chiavi crittografate indelebili e protette a livello hardware

- L'identità del dispositivo è protetta a livello hardware
- Soluzioni hardware per il "Secure Boot"
- Soluzione hardware per l'integrità del sistema

Alcune proprietà dipendono dall'hardware e dal software



Defense in Depth



Dynamic Compartments



Small Trusted Computing Base

Dynamic Compartments

Barriere interne proteggono da ogni singolo problema

- Hardware per creare barriere
- Software per creare sezionamenti controllabili

Alcune proprietà
dipendono dall'hardware,
dal software e dal CLOUD



Certificate-Based
Authentication



Failure
Reporting



Renewable
Security

Renewable Security

Sicurezza del dispositivo sempre aggiornata per superare le minacce in evoluzione

- Il CLOUD per fornire aggiornamenti
- Il software per eseguire gli aggiornamenti
- Hardware per prevenire “ripensamenti”

Ottemperare a queste sette proprietà è difficile e costoso

Disegnare e creare soluzioni olistiche



Riconoscere e mitigare le minacce emergenti



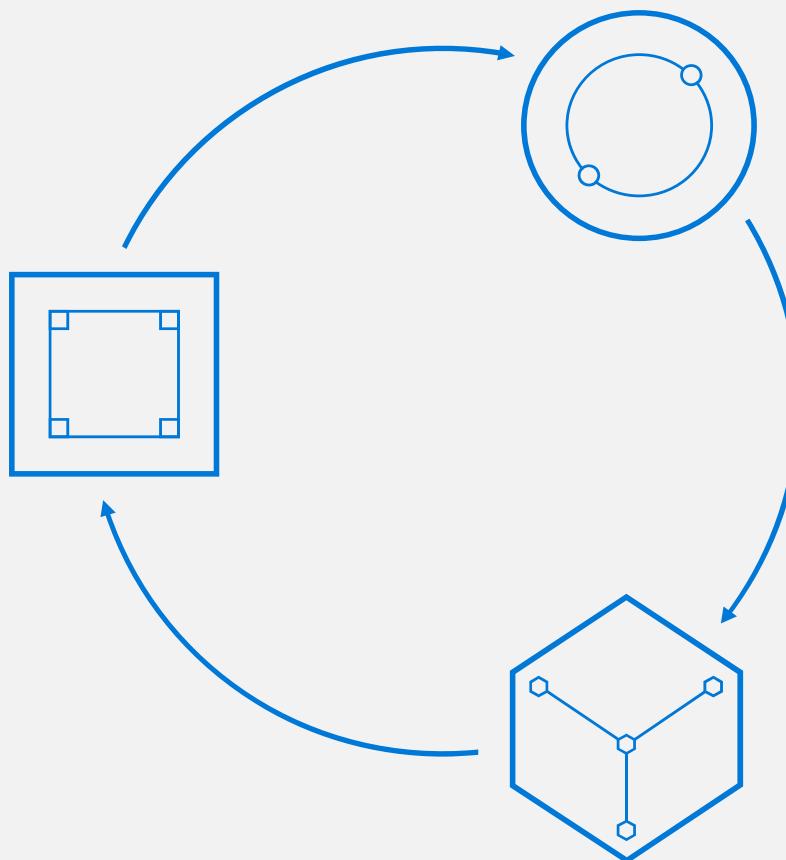
Distribuire e applicare gli aggiornamenti su scala globale



Azure Sphere è una soluzione completa per creare dispositivi con un alto livello di sicurezza

MCU sicura

Una nuova famiglia trasversale di **MCU Azure Sphere** che arriva dai produttori di silicio che, grazie alla tecnologia di sicurezza Microsoft offre: connettività, alte prestazioni e sicurezza a livello hardware.



Sistema Operativo sicuro

L'OS IoT di Azure Sphere ad alta sicurezza combina il meglio delle tecnologie Microsoft e OSS per creare una piattaforma affidabile per nuove esperienze IoT



Protetto dal servizio AZURE

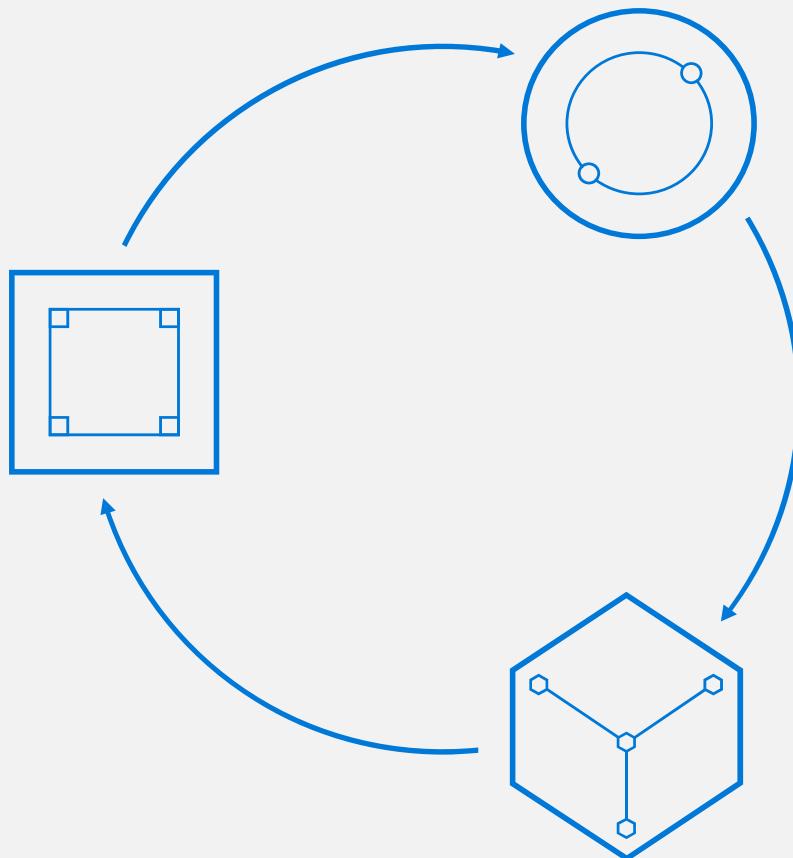
Azure Sphere Security Service protegge ogni dispositivo Azure Sphere; protegge i dispositivi e i clienti, rileva le minacce emergenti e risponde in modo proattivo.



Azure Sphere esploriamo la tecnologia e la soluzione.

Cosa NON è Azure Sphere:

soltanto un hardware
soltanto un Sistema Operativo
soltanto una soluzione CLOUD



Cos'è Azure Sphere:

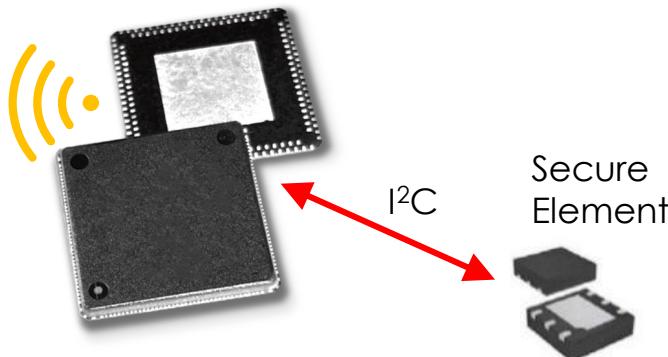
Una soluzione sicura formata da uno speciale MCU, uno speciale sistema Operativo e dai suoi servizi CLOUD.

Oggi l'offerta contiene:

- Chip Azure Sphere → MT3620
- Azure Sphere IoT OS →
Aggiornamenti inclusi fine al 2031
- Servizi di Sicurezza Azure Sphere →
Copertura a vita.

Oggi, Azure Sphere è l'unica soluzione che possiede le sette proprietà per un dispositivo altamente sicuro.

Sviluppo standard per un MCU:



Chip

MT3620

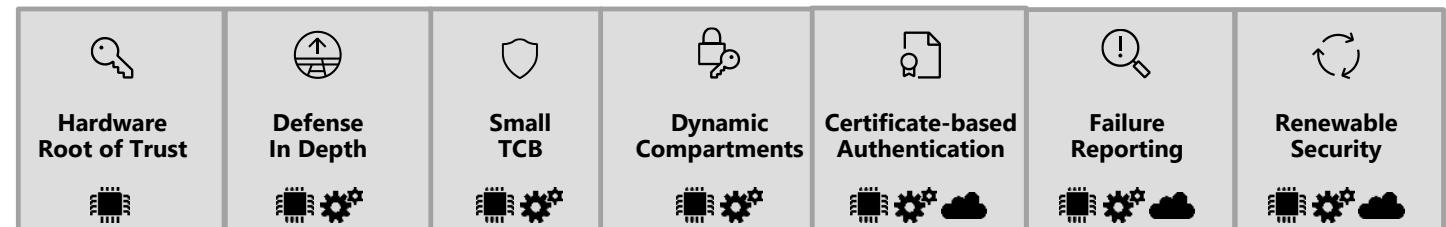
OS & CLOUD Services

AZURE SPHERE



= Supporto Hardware

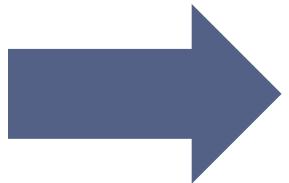
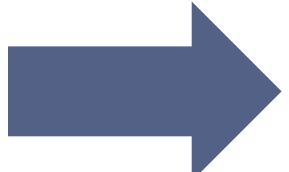
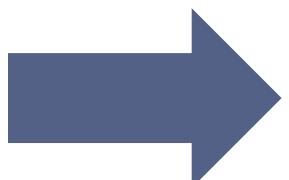
- Vengono forniti esempi di codice, bisogna fare attenzione allo sviluppo
- La “sicurezza” è rinnovabile
- Alterazione dei protocolli? -> Cosa si può fare?
- “HW tampering” come per XBox 360



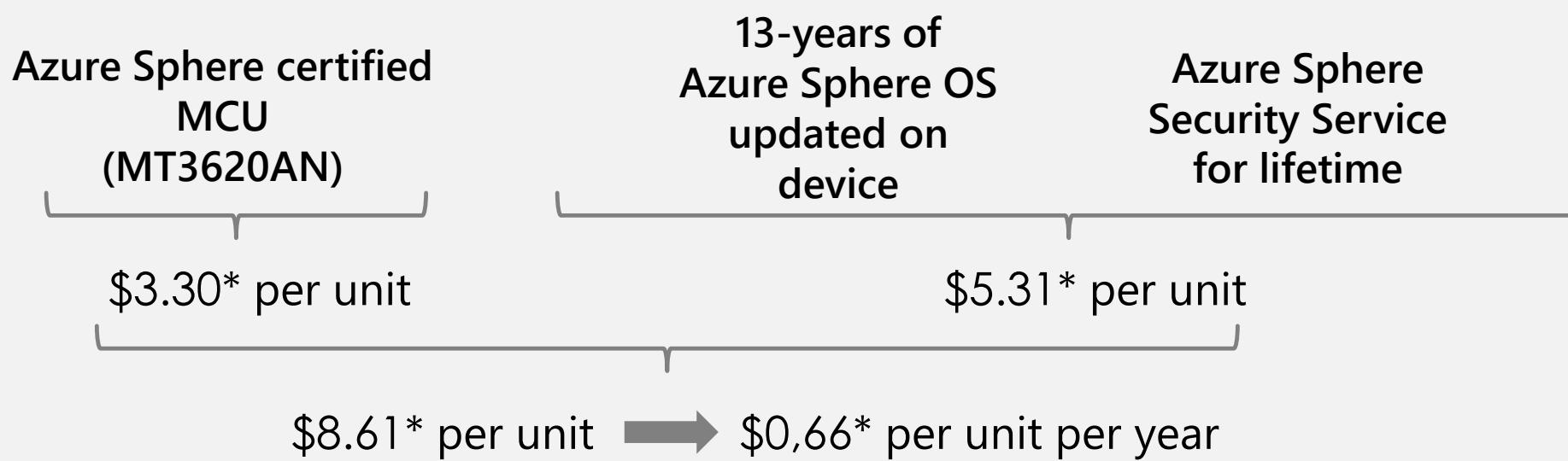
= Supporto Sistema Operativo

= Supporto Servizi CLOUD

Azure Sphere . . . un po' di storia.

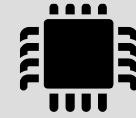
-  **Il progetto nasce nel 2014 in Microsoft Research (ora divisione AI & R)**
Nel 2015 partono i lavori per il prototipo del chip e del sistema operativo
Subito dopo vengono pubblicate le sette proprietà per un dispositivo altamente sicuro (Galen Hunt, George Letey, e Edmund B. Nightingale).
-  **Nel 2017 inizia la «sfida per la sicurezza» basata sul chip e sul sistema operativo**
-  **Dal 2018 è partita una grande collaborazione con i produttori di silicio, i partner e i clienti per la commercializzazione.**

Azure Sphere . . . un'occhiata ai prezzi.

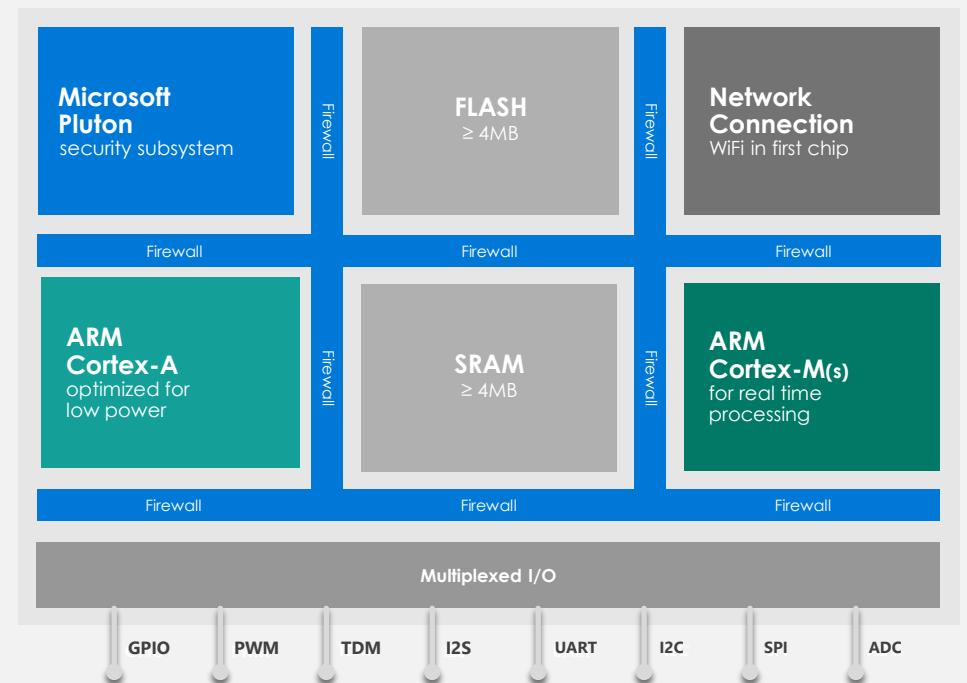


* prices for 1 - 100k quantities. Price will be lower in higher quantities

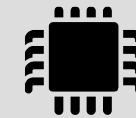
Azure Sphere . . . Il chip MT3620



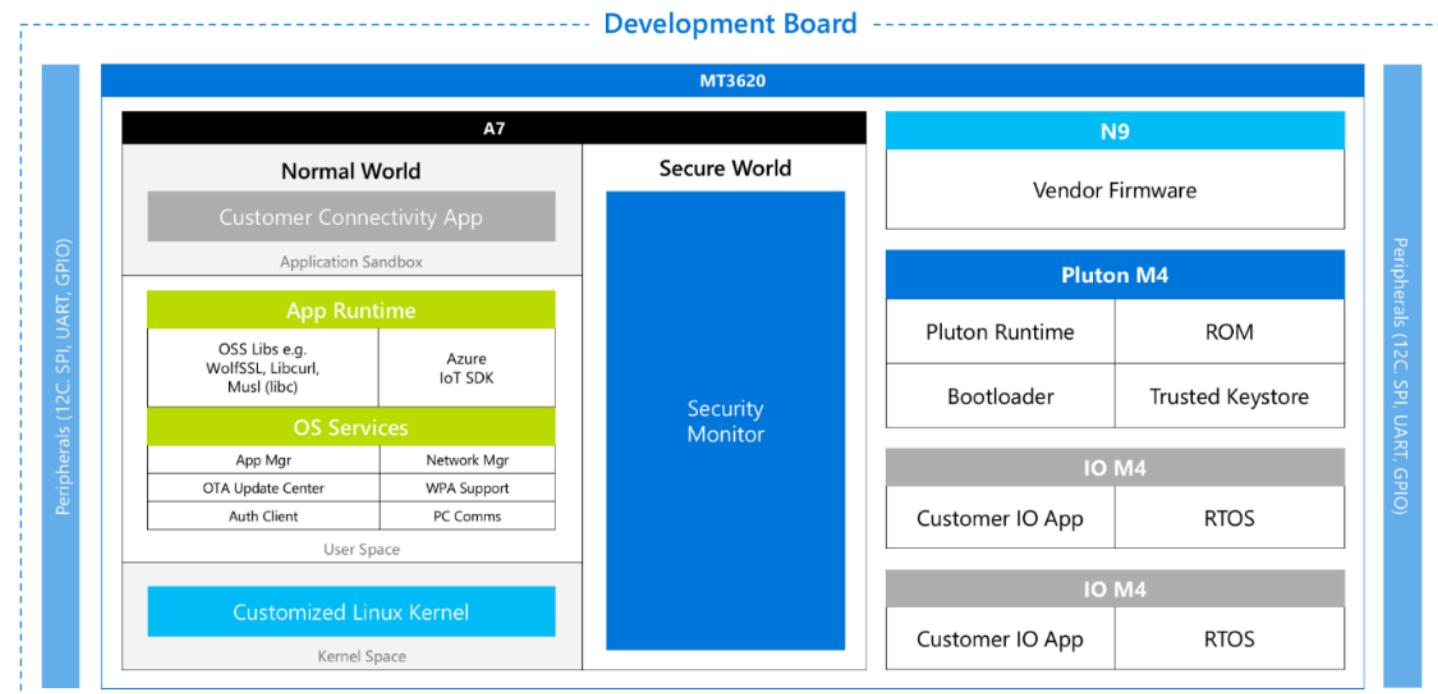
- Sicuro
 - Costruito con le soluzioni tecnologiche di sicurezza di Microsoft (es: I/O bus firewalls tra i sotto-sistemi)
 - Contiene il «Pluton Security Subsystem»
- Performante
 - Contiene 1 Cortex A7 e 2 Cortex M4
 - Ha maggiori performance rispetto ai MCU tradizionali
- Connesso
 - Attraverso una connessione interna



MT3620: MCU certificata Azure Sphere

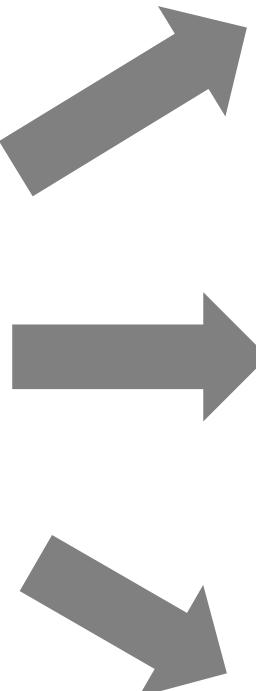
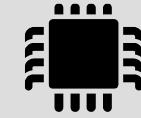


CPUs	ARM Cortex A7 (500MHz) + 2 x Cortex M4 (192MHz)*
RAM	4MB
Flash	16MB in MT3620AN, 32MB in MT3620BN
Connectivity	WiFi 802.11 b/g/n, dual band: 2.4GHz, 5GHz
Microsoft Security	Firewalls, Crypto Accelerator: AES-256, SHA-2, ECC, RSA2K, e-Fused private and public keys, attestation, ...
GPIO	24, 4 configurable as PWM
SPI*	
I/O	6 configurable
UART	
ADC*	8 Channels, 12bit SAR, 2M sample/sec
I2S/TDM*	I2S (2 interfaces) or TDM (4 channels)
Package	DR-QFN 164



* Enabled in MT3620 hardware; software enablement on roadmap

Sviluppo di dispositivi con MT3620



Silicon Partners pronti ad implementare il “Pluton Security Core”



MediaTek



Silicon Labs



Toshiba



ARM



Nordic



VeriSilicon



STMicroelectronics



Nuvoton



NXP



Hilscher

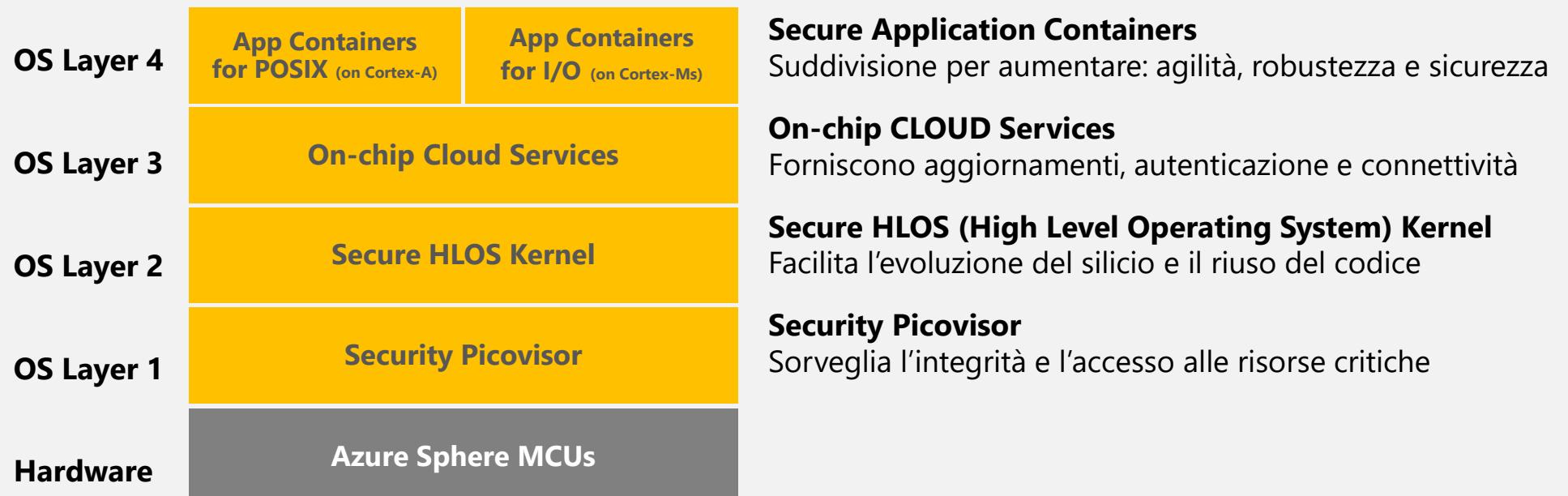


Qualcomm

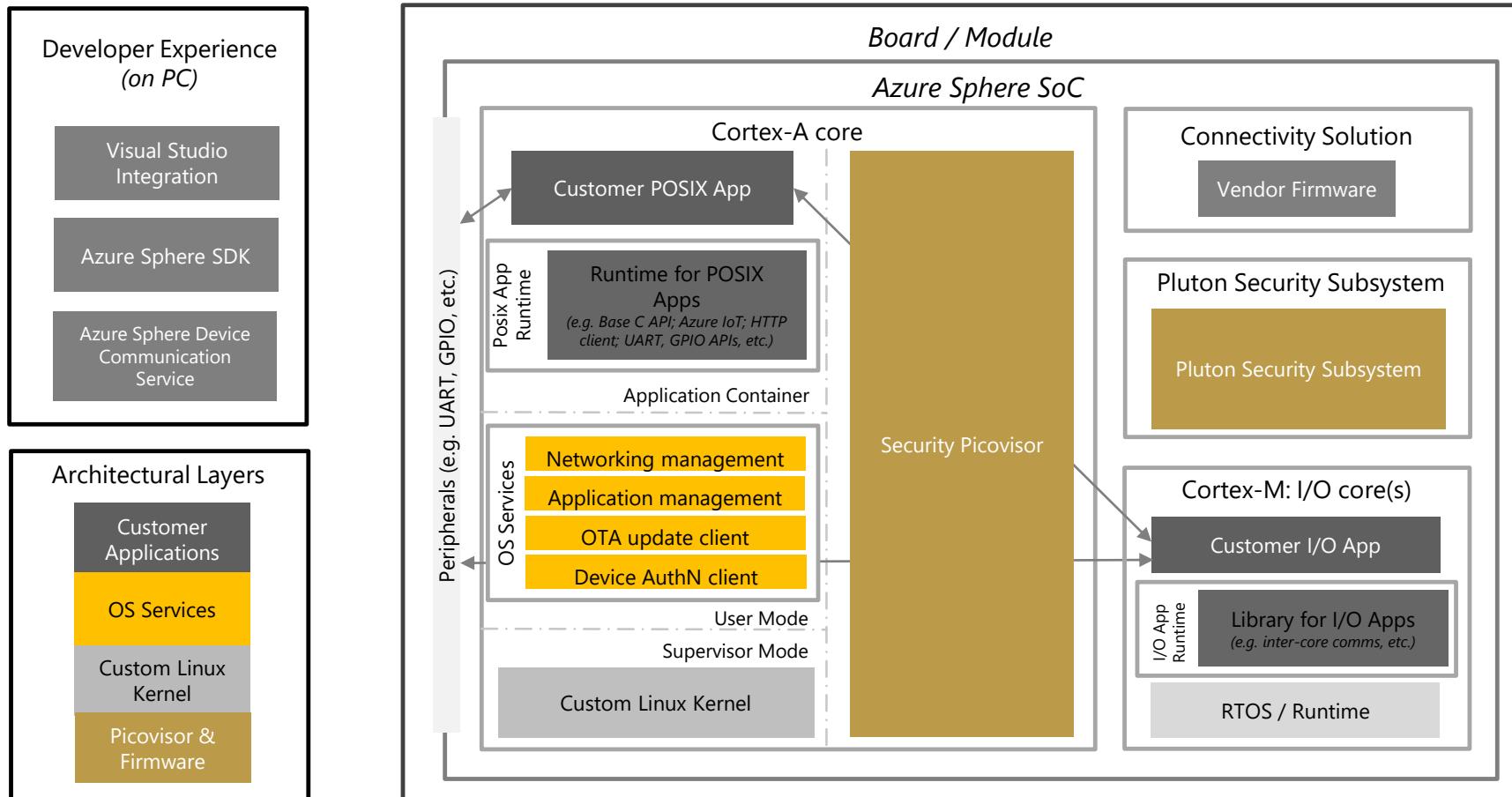
Azure Sphere . . . Il Sistema Operativo



Azure Sphere OS è il primo sistema operativo basato su Linux distribuito da Microsoft



Azure Sphere IoT O.S.: Architettura di base



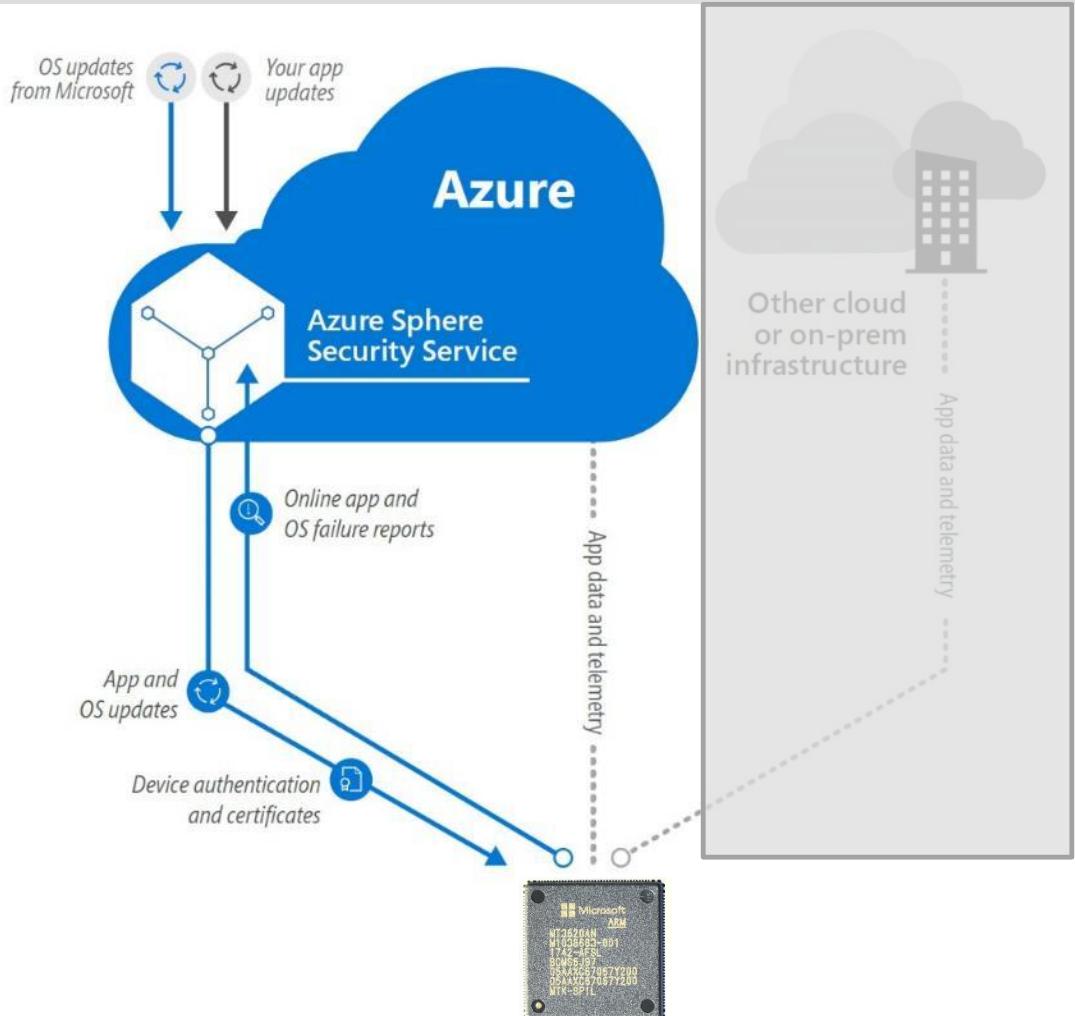
Azure Sphere: i servizi di sicurezza in Azure



Il servizio di sicurezza di Azure Sphere sorveglia ogni Azure Sphere MCU. Gli «rinnova» la sicurezza, identifica situazioni di emergenza e interruzione di sicurezza tra i dispositivi, il cloud e altri «endpoint».

Ecco cosa forniscono:

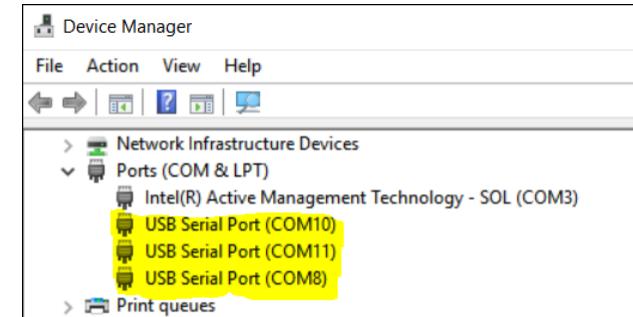
- Protezione dei dispositivi con autenticazioni basate su certificati;
- Garanzia di autenticità del dispositivo con autorizzazione per eseguire soltanto software "genuino"
- Reperimento di informazioni sul dispositivo e sulle applicazioni
- Report di fallimenti e visibilità per le minacce emergenti
- Distribuzione degli aggiornamenti delle applicazioni



Azure Sphere . . . i passi per lo sviluppo

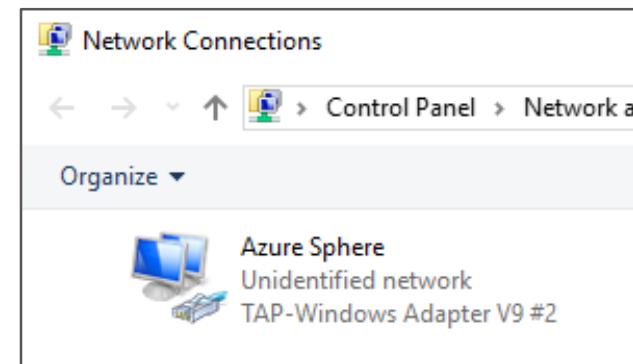
Prerequisiti:

- Avere una scheda di sviluppo Azure Sphere (Development-KIT)
- Un PC con Windows 10 (ver. 1607 o successiva)
- Una porta USB disponibile
- Visual Studio 2017 (ver. 15.7 o successiva)



Connessione della scheda

Quando si connette la scheda via USB compariranno 3 nuove COM



Installazione del SDK

Ecco il link per il download:

[Download the Azure Sphere SDK Preview for Visual Studio](#)

Con l'SDK viene creato:

- Un «trasporto di rete dedicato»;
- Un applicativo console: azsphere.exe.

```
F:\Users\beppe\Documents>azsphere
Missing required positional argument '<Operation>'.
Azure Sphere Utility version 19.2.4.2632
Copyright (C) Microsoft Corporation. All rights reserved.

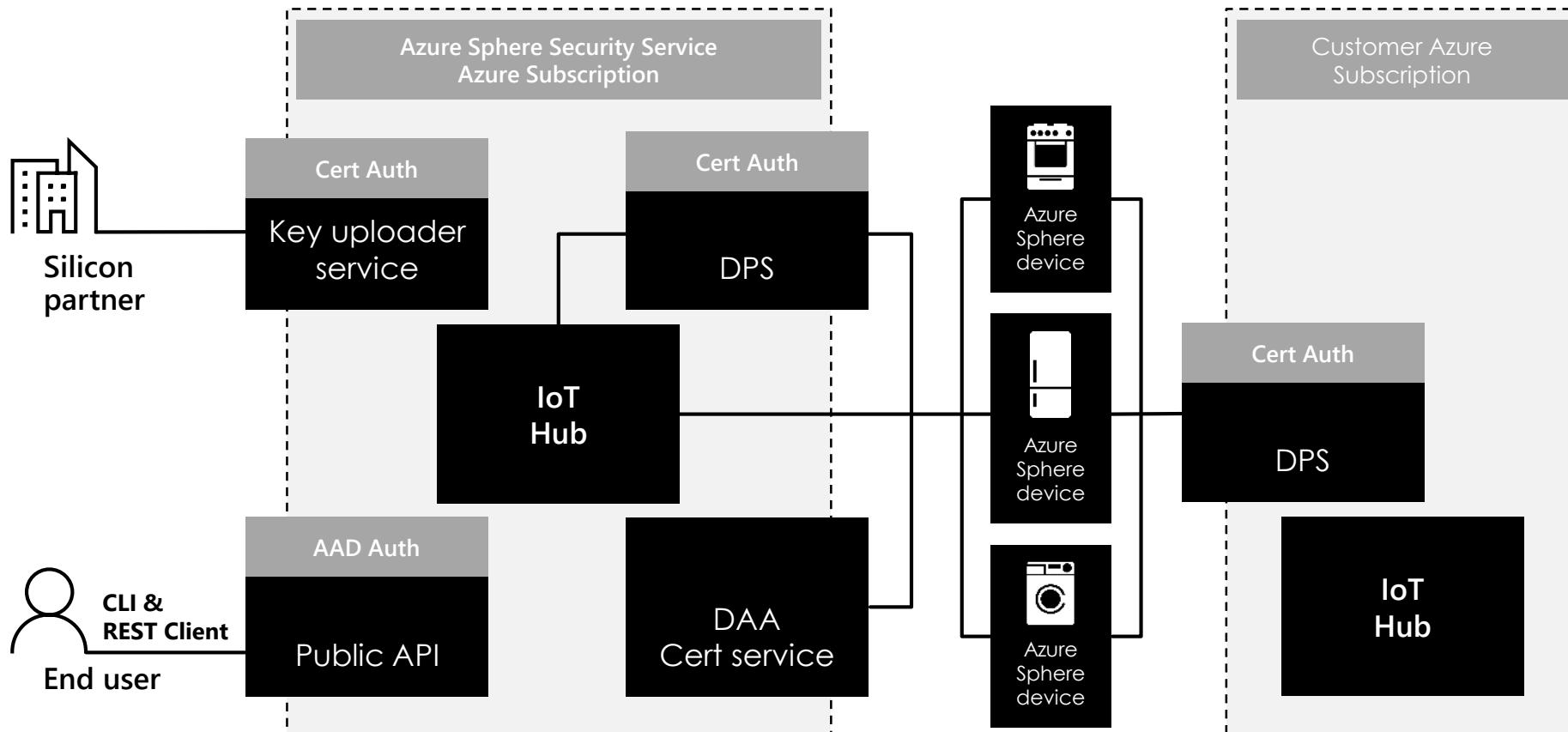
Azure Sphere command line tool: azsphere.exe

Required Parameters:
<Operation>
    com_component - Manage components and images in your Azure Sphere tenant.
    dev_device - Manage devices.
    dg_device_group - Manage device groups in your Azure Sphere tenant.
    feed - Manage feeds in your Azure Sphere tenant.
    get-support-data - Gather diagnostic data about your system, cloud and device configurations.
    img_image - Manage image packages.
    ins_image-set - Manage image sets in your Azure Sphere tenant.
    login - Log in to the Azure Sphere Security Service.
    log-out - Log out from the Azure Sphere Security Service.
    show-version - Show the version of the Azure Sphere tools.
    sku - Manage SKUs in your Azure Sphere tenant.
    tenant - Manage Azure Sphere tenants.

Optional Parameters:
--help - Show help information.
--verbose - Show additional information.

F:\Users\beppe\Documents>
```

Azure Sphere: i servizi di sicurezza e di IoT in Azure



Azure Sphere: abbinamento del dispositivo (Tenant Claim)



Ogni MCU di Azure Sphere deve essere conosciuta da Azure Active Directory (AAD) di proprietà dell'azienda. Cioè ogni scheda di sviluppo, prima di essere utilizzata, deve essere abbinata al nostro account aziendale AAD.

L'account aziendale AAD deve essere configurato da Microsoft per lavorare con Azure Sphere.

Microsoft usa il portale "MS Collaborate" per comunicare e abilitare gli AAD per Azure Sphere. Un amministratore dell'azienda OEM può definire nuovi AAD e generare utenti per Azure Sphere e utilizzare il portale per fornire feedback diretti, porre domande, segnalare bug e generare nuove richieste di funzionalità.

Ecco come fare:

1. Assicuratevi di essere registrati nel Microsoft Developer Center Account (se ne siete già provvisti passate al punto 3)
2. Ecco il link con le istruzioni per registrarsi al Microsoft Developer Center Account:
<https://docs.microsoft.com/en-us/windows/uwp/publish/opening-a-developer-account#additional-guidelines-for-company-accounts>
3. Ecco il link per registrarsi al programma Microsoft Collaborate:
<https://docs.microsoft.com/en-us/collaborate/registration>

Azure Sphere . . . L'ambiente di sviluppo

Lo sviluppo con Visual Studio migliora la produttività

Semplifica lo sviluppo

Ci si può concentrare sul proprio obiettivo

Accelerata la distribuzione

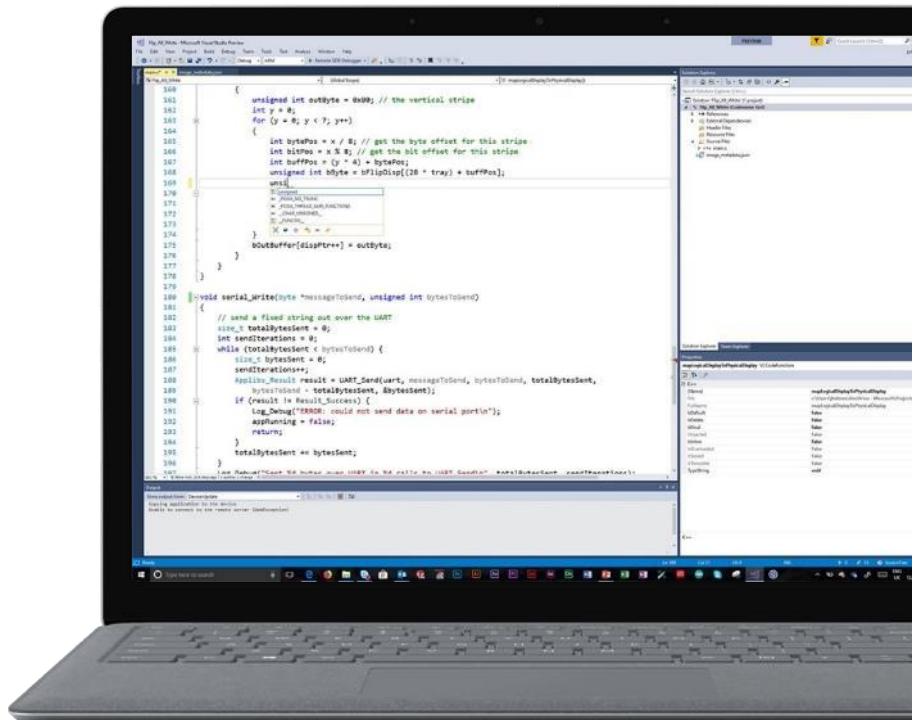
Integrandola nell'ambiente di sviluppo

Snellisce il debugging

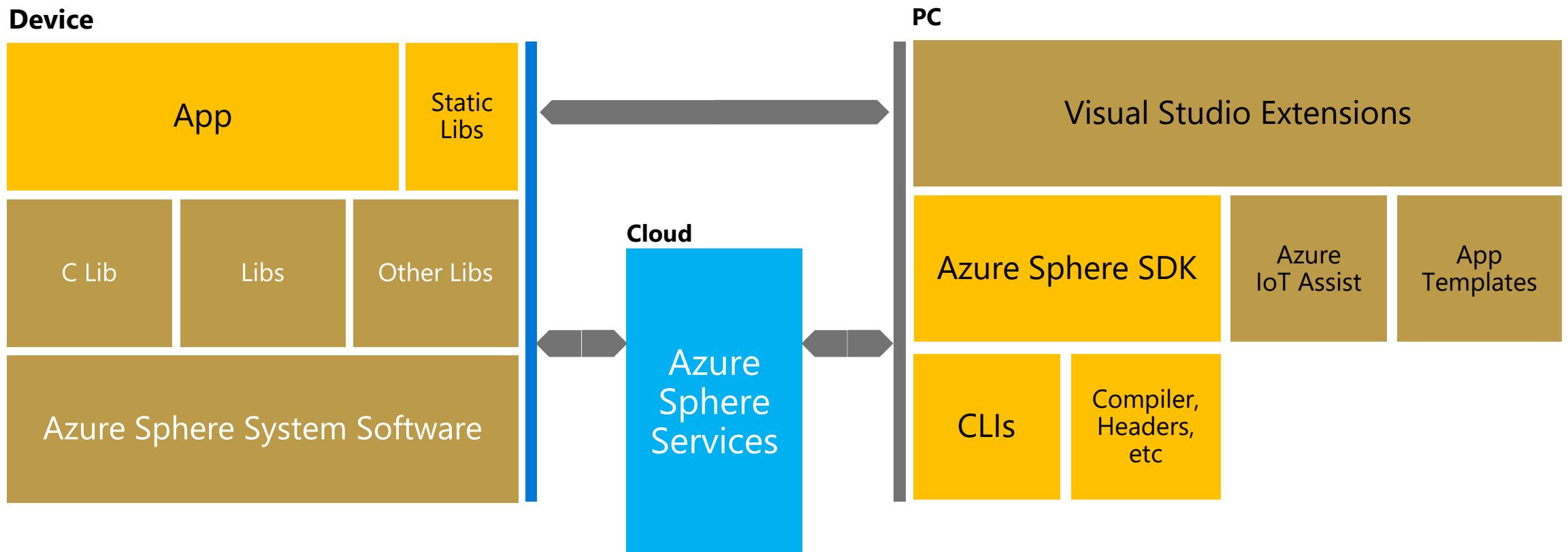
Debug remoto interattivo sul dispositivo (e sul cloud)

Facilita la collaborazione nel team di sviluppo

Con l'accesso ai tool di collaborazione (GitHub, DevOps, TEAM)



Azure Sphere . . . Architettura per lo sviluppo

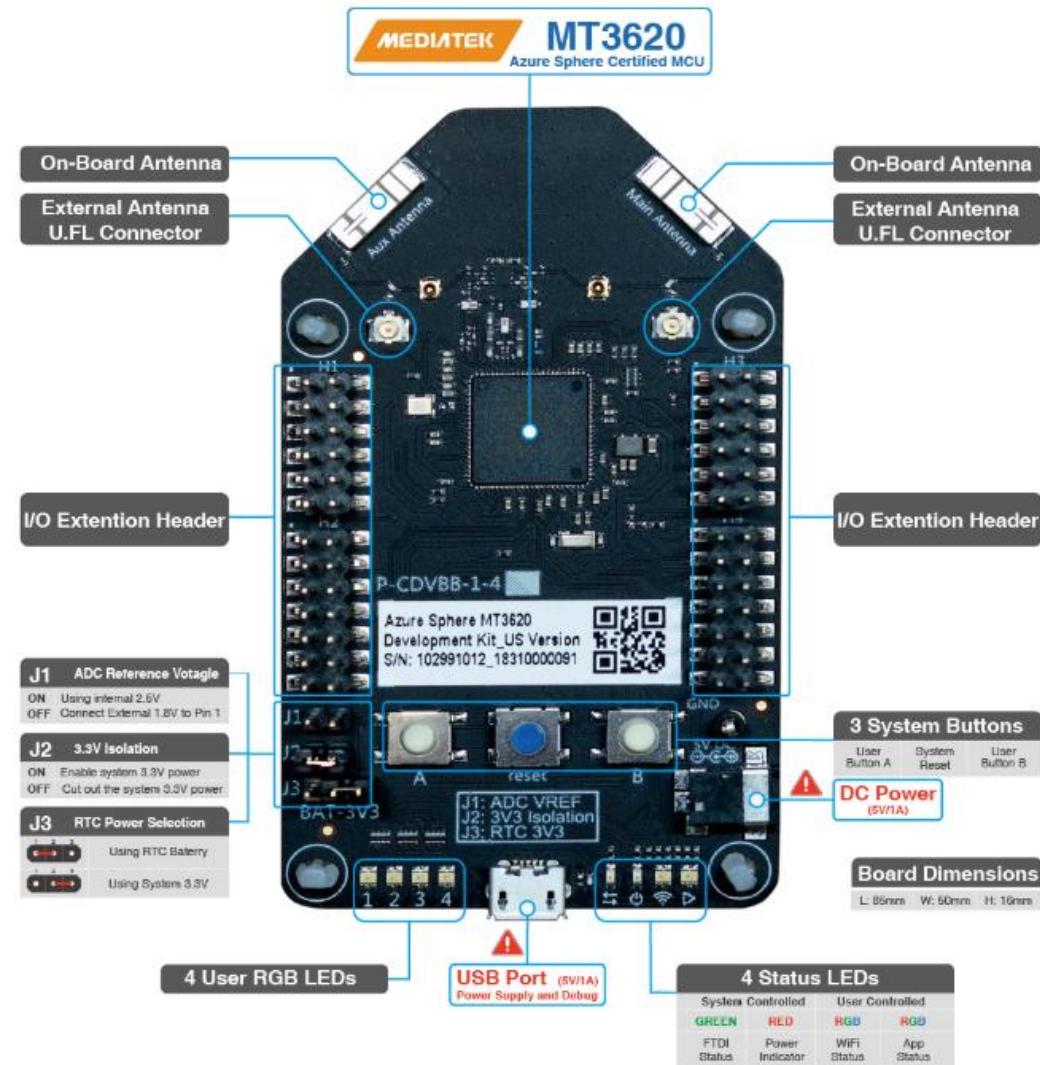
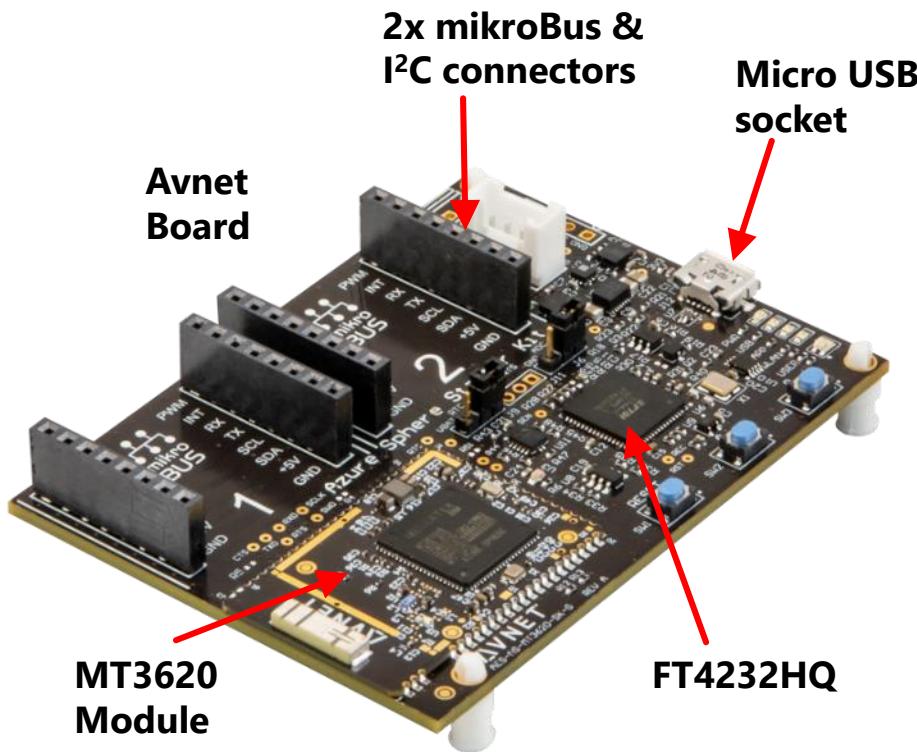


Hello Sphere . . . L'ambiente di sviluppo

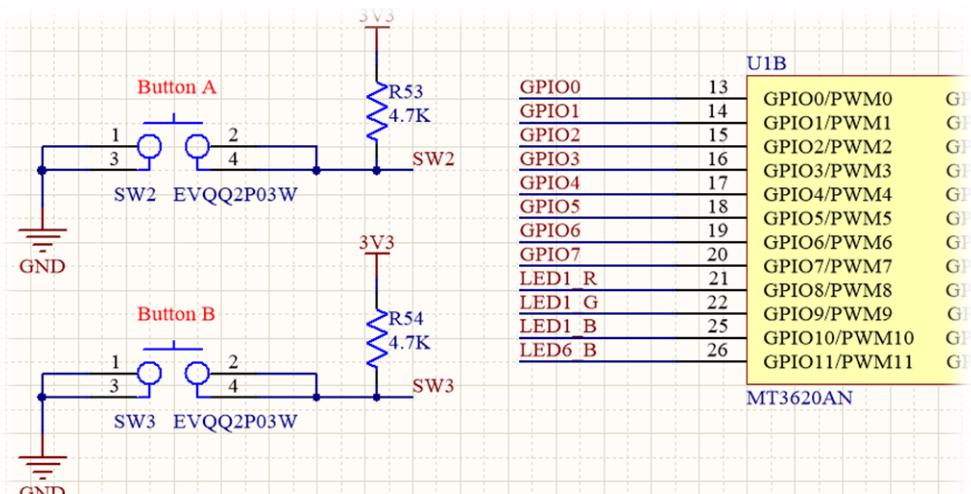
The screenshot shows the Microsoft Visual Studio interface for developing applications for the Azure Sphere platform. The main window displays the code for `main.c` in the `Mt3620Blink1` project. The code is a C program that initializes peripherals and handles events using epoll. It includes comments explaining the main entry point and the event loop. The Visual Studio interface includes the Solution Explorer, Cloud Explorer, and various toolbars and status bars.

```
175 ///// <summary>
176 ///// Main entry point for this application.
177 ///// </summary>
178 int main(int argc, char *argv[])
179 {
180     Log_Debug("Blink application starting.\n");
181     if (InitPeripheralsAndHandlers() != 0) {
182         terminationRequired = true;
183     }
184
185     // Use epoll to wait for events and trigger handlers, until an error or SIGTERM happens
186     while (!terminationRequired) {
187         if (WaitForEventAndCallHandler(epollFd) != 0) {
188             terminationRequired = true;
189         }
190     }
191
192     ClosePeripheralsAndHandlers();
193     Log_Debug("Application exiting.\n");
194     return 0;
195 }
196
```

Azure Sphere MT3620 - Development Kit

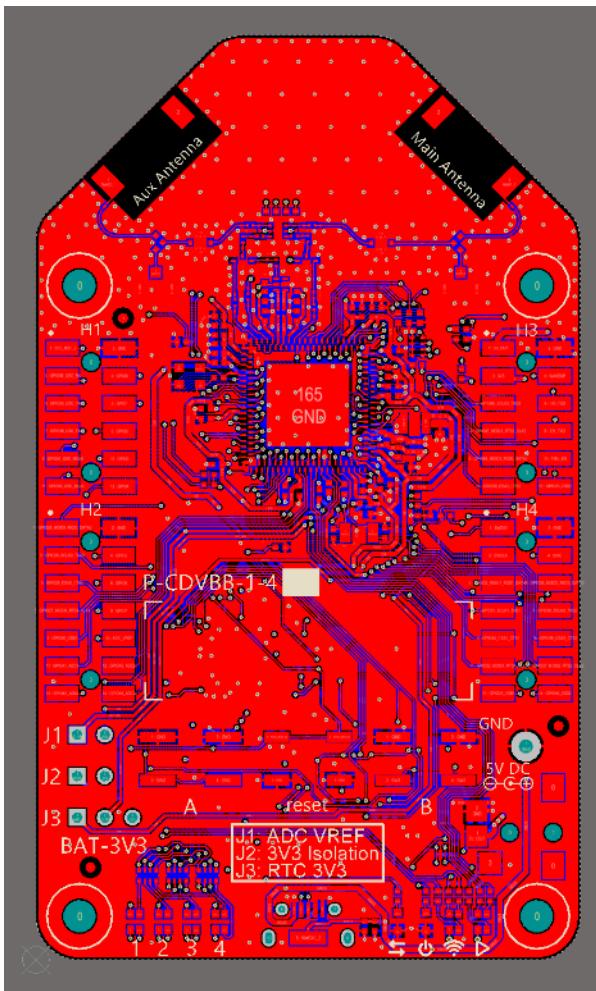


Azure Sphere Dev-Kit – Open Source Reference Design

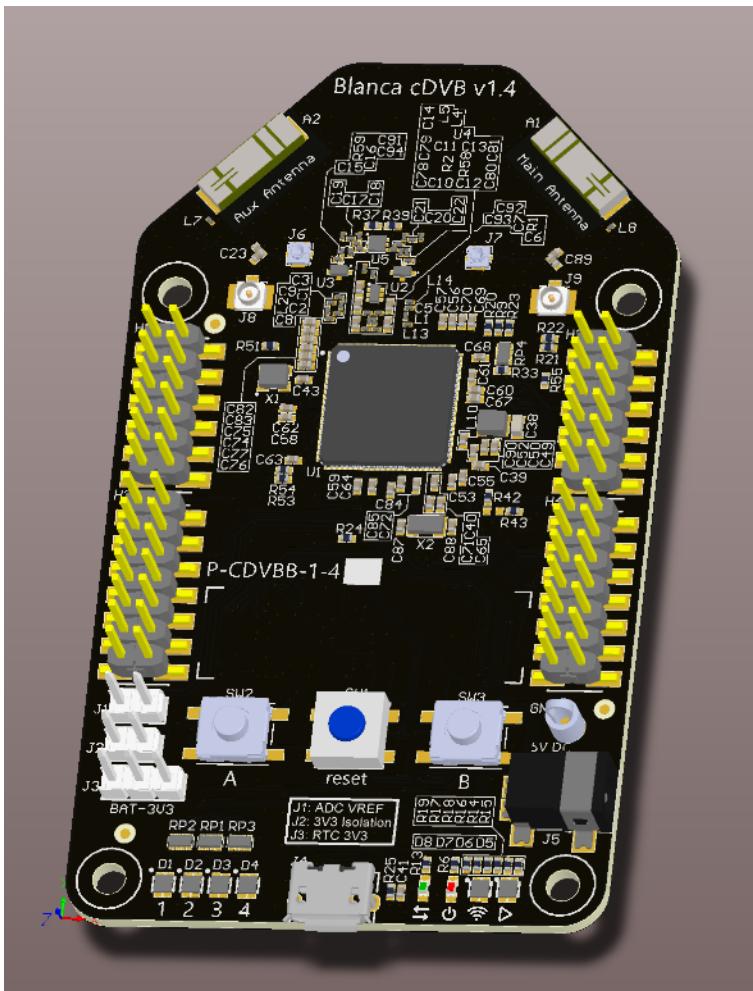


U1J	
GPIO26_SCLK0_TXD0	39
GPIO27_MOSI0 RTS0 CLK0	40
GPIO28_MISO0 RXD0 DATA0	42
GPIO29_CSA0 CTS0	43
GPIO30_CSB0	45
GPIO31_SCLK1 TXD1	46
GPIO32_MOSI1 RTS1 CLK1	47
GPIO33_MISO1 RXD1 DATA1	48
GPIO34_CSA1 CTS1	49
GPIO35_CSB1	50
GPIO36_SCLK2 TXD2	51

Schematic



PCB layout



3D CAD models 38

Azure Sphere Dev-Kit – Dove ordinare

- Da Seeed:

<https://www.seeedstudio.com/Azure-Sphere-MT3620-Development-Kit-EU-Version-p-3134.html>

- Da Avnet:

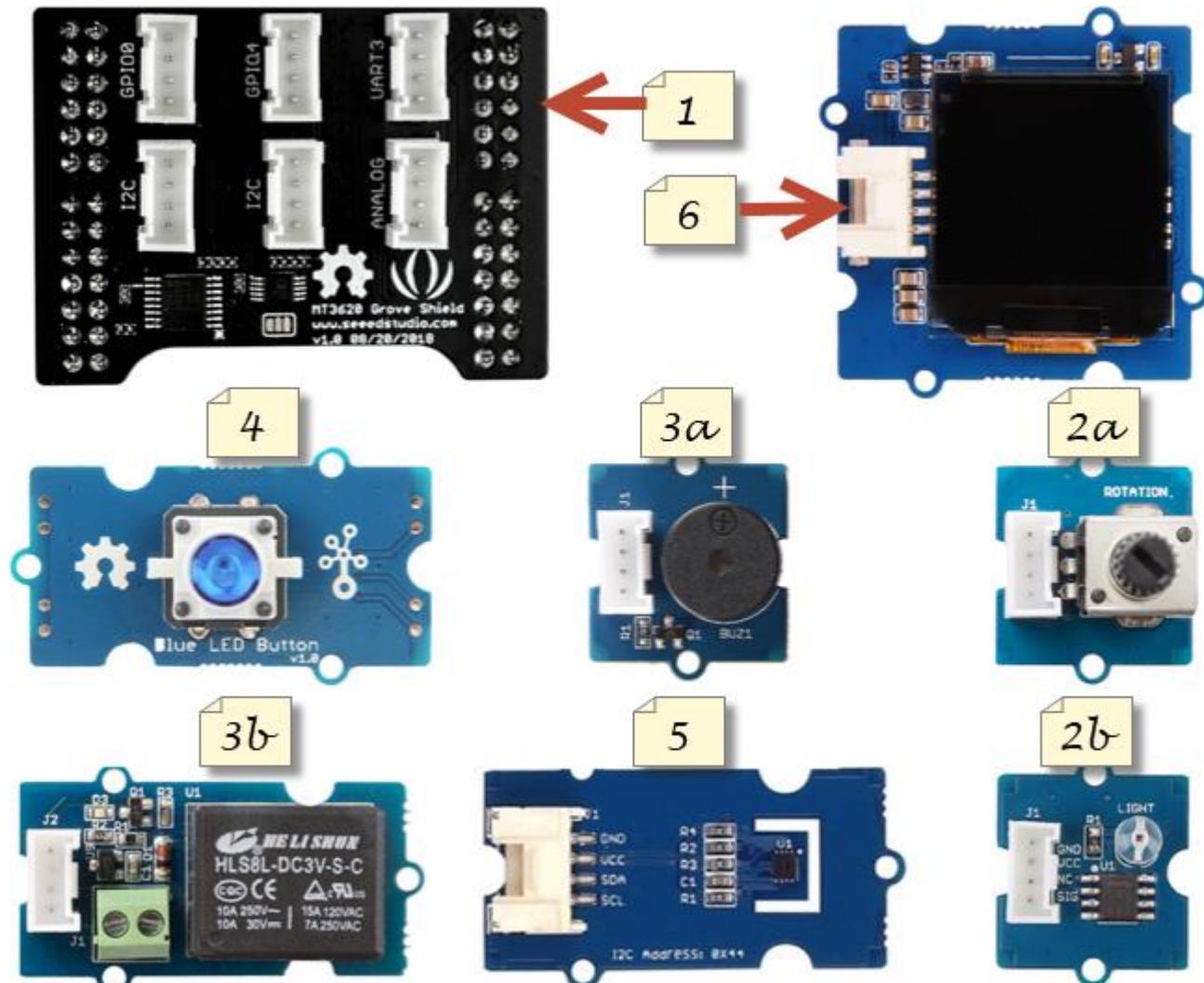
<https://www.avnet.com/shop/us/products/avid-technologies/aes-ms-mt3620-sk-g-3074457345636825680/>

- Prezzo indicativo ~ 75 - 85 USD

Azure Sphere Dev-Kit – Starter Kit

Grove Starter Kit for Azure Sphere MT3620
Development Kit (Prezzo indicativo - 50 USD)

- 1) Shield (di raccordo tra la board e i device);
- 2) Analog Input:
 - a) Sensore di rotazione;
 - b) Sensore di luminosità;
- 3) Digital Input:
 - a) Buzzer;
 - b) Relé;
- 4) Digital I/O (Bottone a LED Blu);
- 5) Sensore di umidità e temperatura;
- 6) OLED Display 1.12"



http://wiki.seeedstudio.com/Grove_Starter_Kit_for_Azure_Sphere_MT3620_Development_Kit/



DEMO

Hello Sphere = Blinking LED

41

Demo: Hello Sphere – passo per passo

```
azsphere
Missing required positional argument '<Operation>'.

Azure Sphere Utility version 19.2.4.2632
Copyright (C) Microsoft Corporation. All rights reserved.

    Azure Sphere command line tool: azsphere.exe

Required Parameters:
<Operation>
    com, component      - Manage components and images in your Azure Sphere tenant.
    dev, device         - Manage devices.
    dg, device-group   - Manage device groups in your Azure Sphere tenant.
    feed                - Manage feeds in your Azure Sphere tenant.
    get-support-data   - Gather diagnostic data about your system, cloud and device configurations.
    img, image          - Manage image packages.
    ims, image-set      - Manage image sets in your Azure Sphere tenant.
    login               - Log in to the Azure Sphere Security Service.
    logout              - Log out from the Azure Sphere Security Service.
    show-version        - Show the version of the Azure Sphere tools.
    sku                 - Manage SKUs in your Azure Sphere tenant.
    tenant              - Manage Azure Sphere tenants.

Optional Parameters:
-?, --help            - Show help information.
-v, --verbose          - Show additional information.

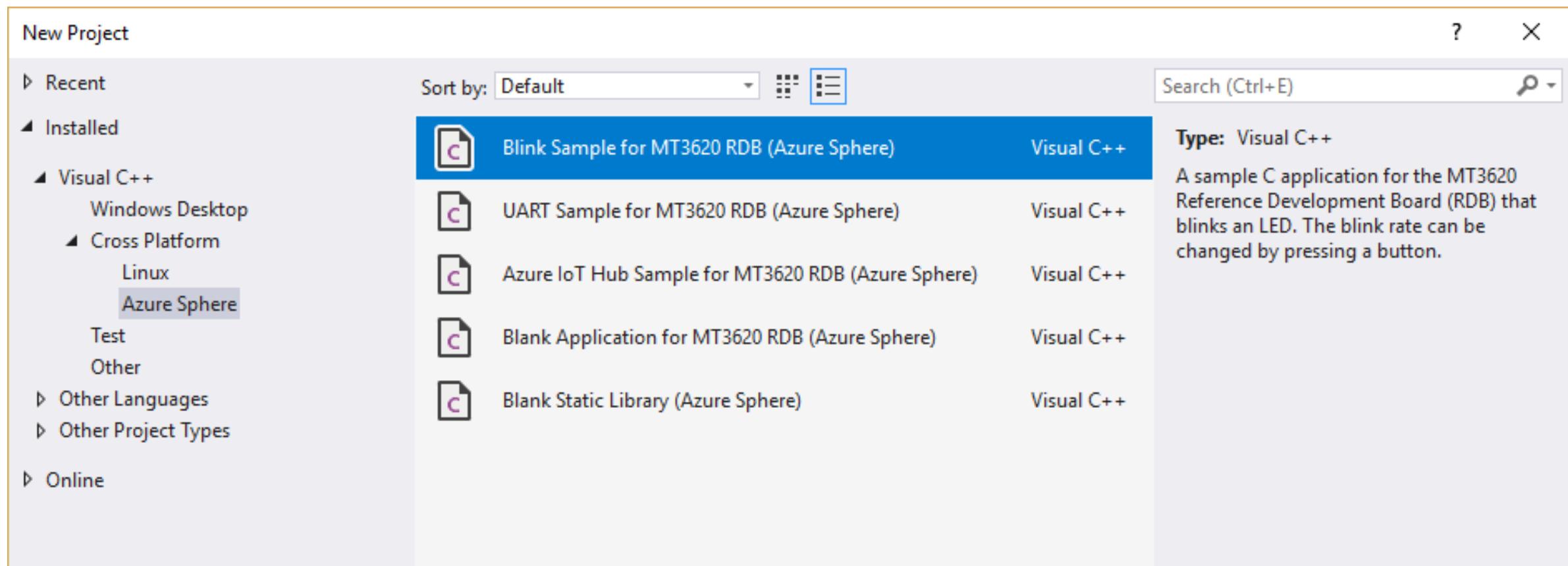
F:\Users\beppe\Documents>
```

Demo: Hello Sphere – passo per passo

```
azsphere device prep-debug
Getting device capability configuration for application development.
Downloading device capability configuration for device ID 'B76F488136B7D7BD8AFB4975F778263CAD8E
E4154930412689E2BC2ED122818B3A5287D648190ED3B8956AB38AB65B020EDD4605C6BA5D073908C5E19ACBC81D'.
Successfully downloaded device capability configuration.
Successfully wrote device capability configuration file 'C:\Users\beppe\AppData\Local\Temp\tmp4
95B.tmp'.
Setting device group ID 'cd037ae5-27ca-4a13-9e3b-2a9d87f9d7bd' for device with ID 'B76F488136B7
D7BD8AFB4975F778263CAD8EE4154930412689E2BC2ED122818B3A5287D648190ED3B8956AB38AB65B020EDD4605C6B
A5D073908C5E19ACBC81D'.
Successfully disabled over-the-air updates.
Enabling application development capability on attached device.
Applying device capability configuration to device.
Successfully applied device capability configuration to device.
The device is rebooting.
Installing debugging server to device.
Deploying 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepackage'
to the attached device.
Image package 'C:\Program Files (x86)\Microsoft Azure Sphere SDK\DebugTools\gdbserver.imagepack
age' has been deployed to the attached device.
Application development capability enabled.
Successfully set up device 'B76F488136B7D7BD8AFB4975F778263CAD8EE4154930412689E2BC2ED122818B3A5
287D648190ED3B8956AB38AB65B020EDD4605C6BA5D073908C5E19ACBC81D' for application development, and
disabled over-the-air updates.
Command completed successfully in 00:00:46.3620150.
```

43

Demo: Hello Sphere – nuovo progetto



Demo: Hello Sphere – remote debug

The screenshot shows a debugger interface with the following details:

- Code View:** Displays a portion of the C code for a button timer application. A red breakpoint icon is visible on line 93. The code includes logic for button presses, GPIO value changes, and event handlers.
- Variables View (Autos):** Shows the current values of three variables:

Name	Type	Value
GPIO_Value_Low	enum { ... }	GPIO_Value_Low
buttonState	GPIO_Value_Type	1 '\001'
newButtonState	GPIO_Value_Type	0 '\000'
- Output View:** Displays logs from the remote debugging session:

```
Show output from: Device Output
Remote debugging from host 192.168.35.1
Blink application starting.
Opening MT3620_RDB_BUTTON_A as input.
Opening MT3620_RDB_LED1_RED.
```



DEMO

IoT HUB + Sphere

46

Demo: IoT-HuB + Azure Sphere – Device Explorer Twin

Ecco da dove scaricare l'eseguibile già pronto:

<https://github.com/Azure/azure-iot-sdk-csharp/releases/tag/2019-1-4>

Azure / azure-iot-sdk-csharp

Code Issues 134 Pull requests 8 Projects 4 Wiki Insights

Watch 98 Star 237 Fork 357

Releases Tags

2019-1-4 · 4f1e5f9

Microsoft Azure IoT Hub SDK for C# Release 2019-1-4

CIPop released this on Jan 5 · 41 commits to master since this release

Microsoft.Azure.Devices.Client v1.19.0

Demo: IoT-HuB + Azure Sphere – Device Explorer Twin

Device Explorer Twin

Configuration Management Data Messages To Device Call Method on Device

Connection Information

IoT Hub Connection String:

HostName=BepsSphereIoTHub.azure-
[REDACTED]

Protocol Gateway HostName:

[REDACTED]

Update

Shared Access Signature

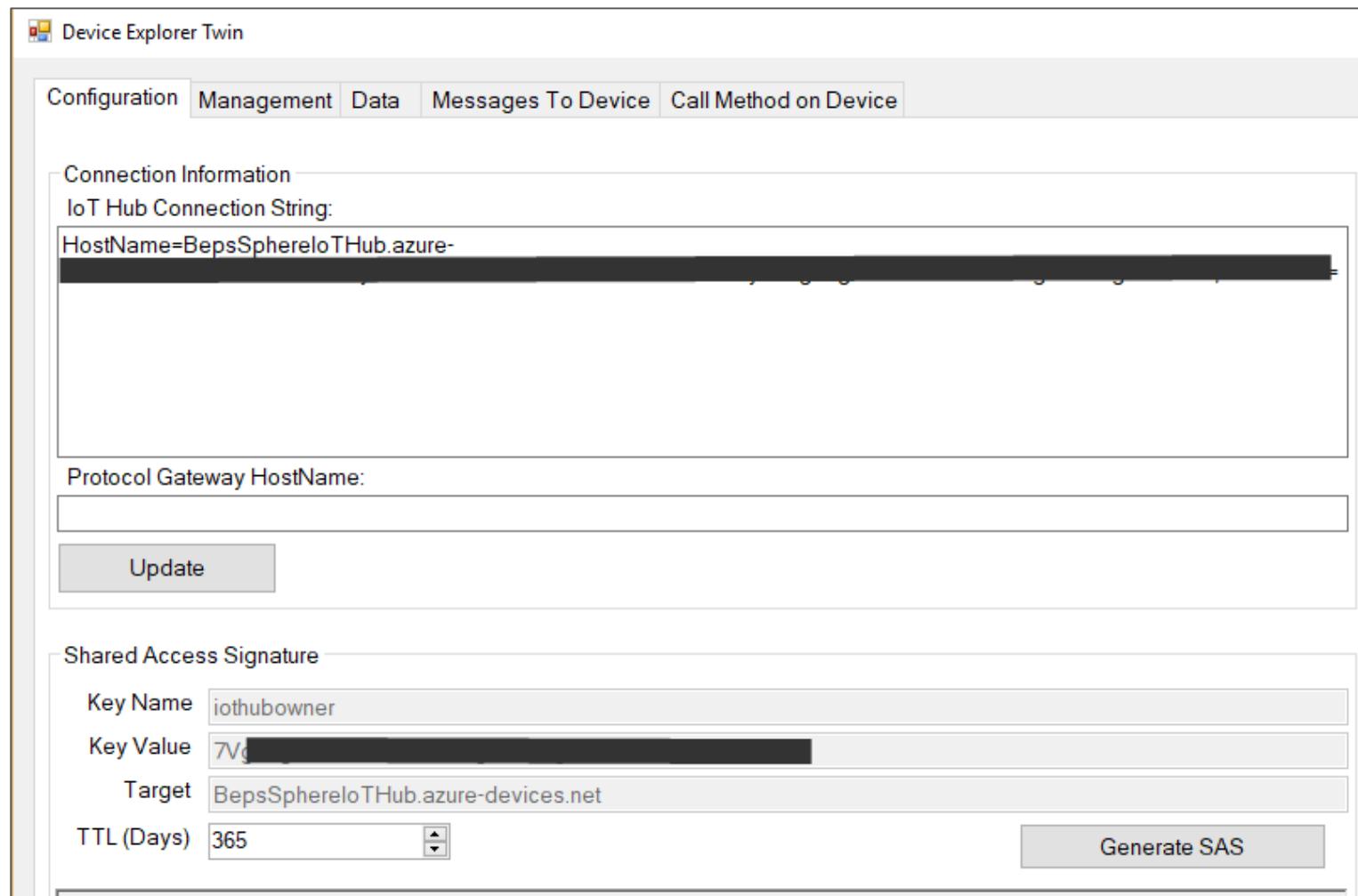
Key Name: iothubowner

Key Value: 7Vc [REDACTED]

Target: BepsSphereIoTHub.azure-devices.net

TTL (Days): 365

Generate SAS



Demo: IoT-HuB + Azure Sphere – Shared access keys

Submit MVP Activity

Microsoft Azure Search resources, services, and docs

Home > All resources > BepsSphereIoTHub - Shared access policies

All resources BEPS Engineering S.r.l.

Add Edit columns More

Filter by name...

NAME
bepsblobsphere
BepsIoTHubService
BepsSphereIoTHub
SphereStreamAnalytics

BepsSphereIoTHub - Shared access policies

IoT Hub

Search (Ctrl+)

Add

Overview

Activity log

Access control (IAM)

Tags

Events

Settings

Shared access policies

Pricing and scale

IP Filter

Certificates

Built-in endpoints

Properties

Locks

Export template

IoT Hub uses permissions to grant access to each IoT hub endpoint based on functionality.

Search to filter items...

POLICY	PERMISSIONS
iothubowner	registry write, service connect, device connect
service	service connect
device	device connect
registryRead	registry read
registryReadWrite	registry write

Access policy name: iothubowner

Permissions:

- Registry read
- Registry write
- Service connect
- Device connect

Shared access keys:

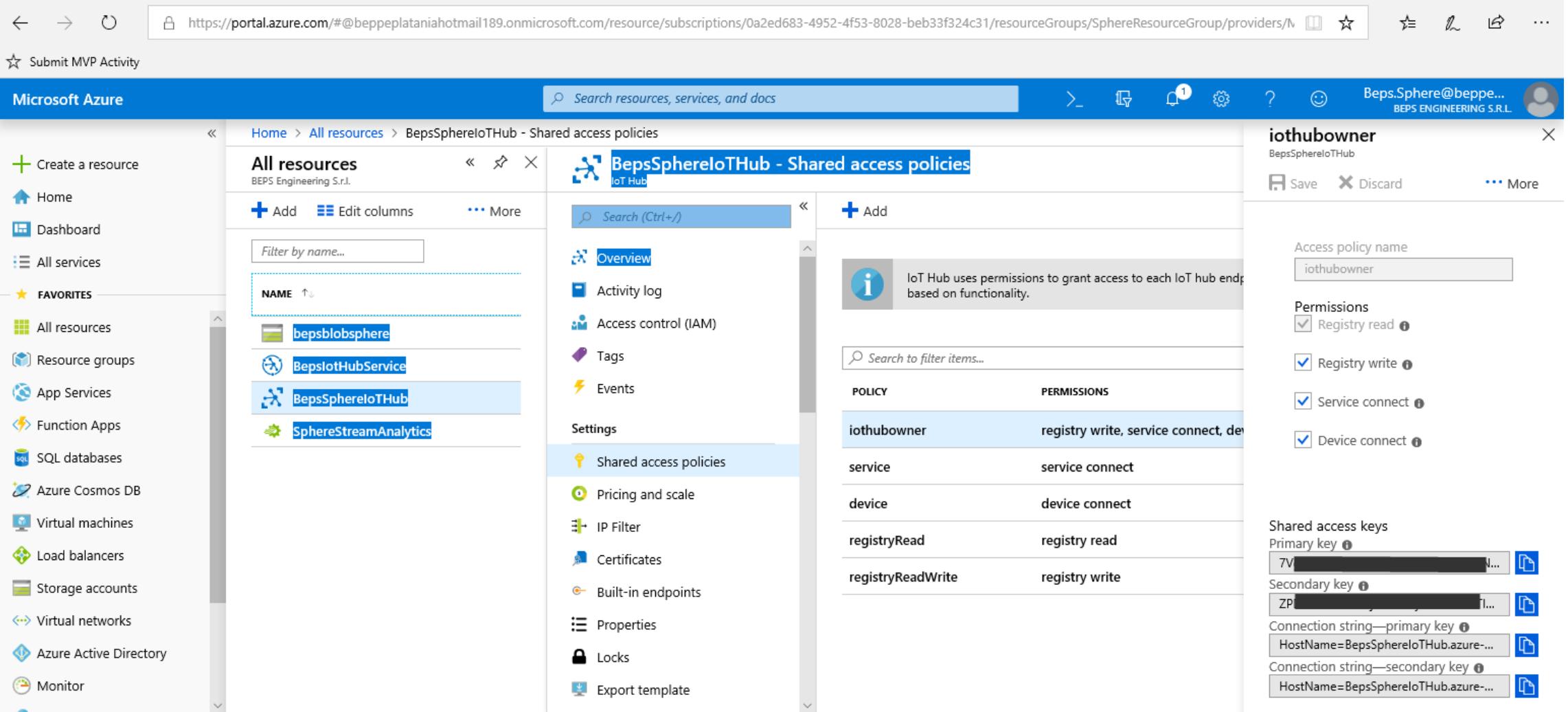
Primary key: 7V [REDACTED] [Copy](#)

Secondary key: ZP [REDACTED] [Copy](#)

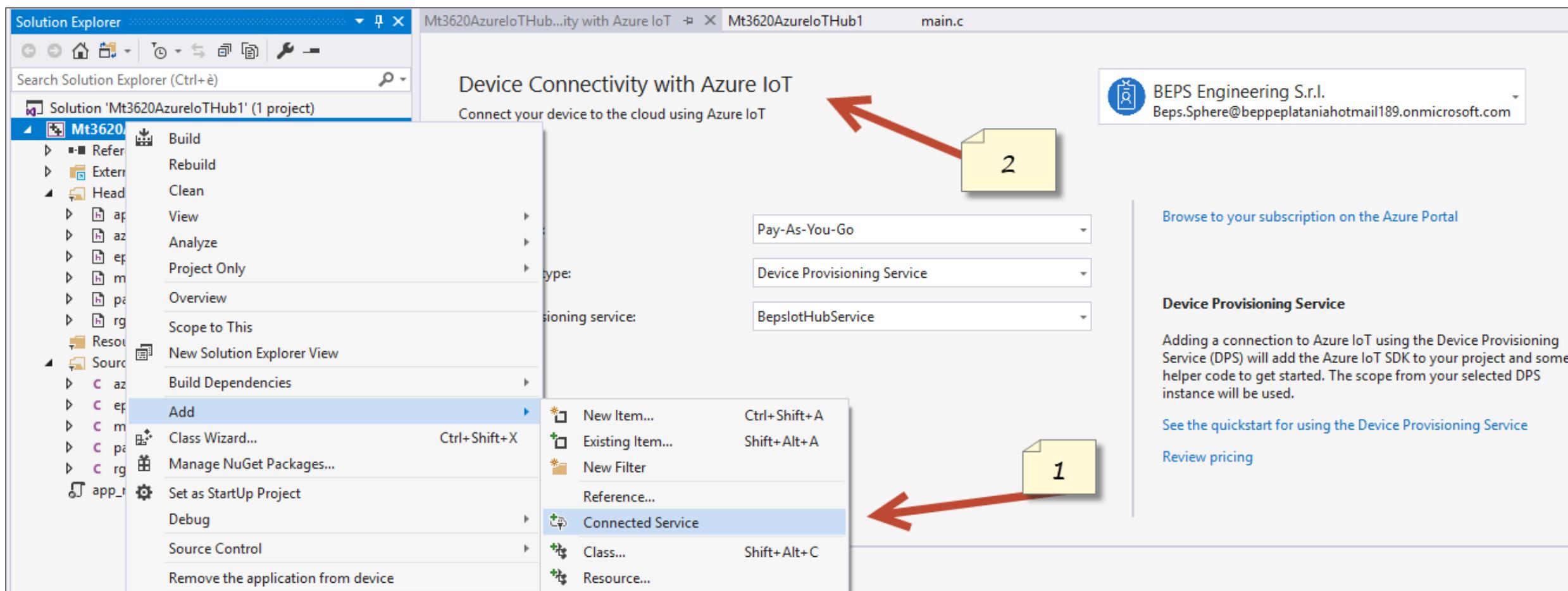
Connection string—primary key: HostName=BepsSphereIoTHub.azure... [Copy](#)

Connection string—secondary key: HostName=BepsSphereIoTHub.azure... [Copy](#)

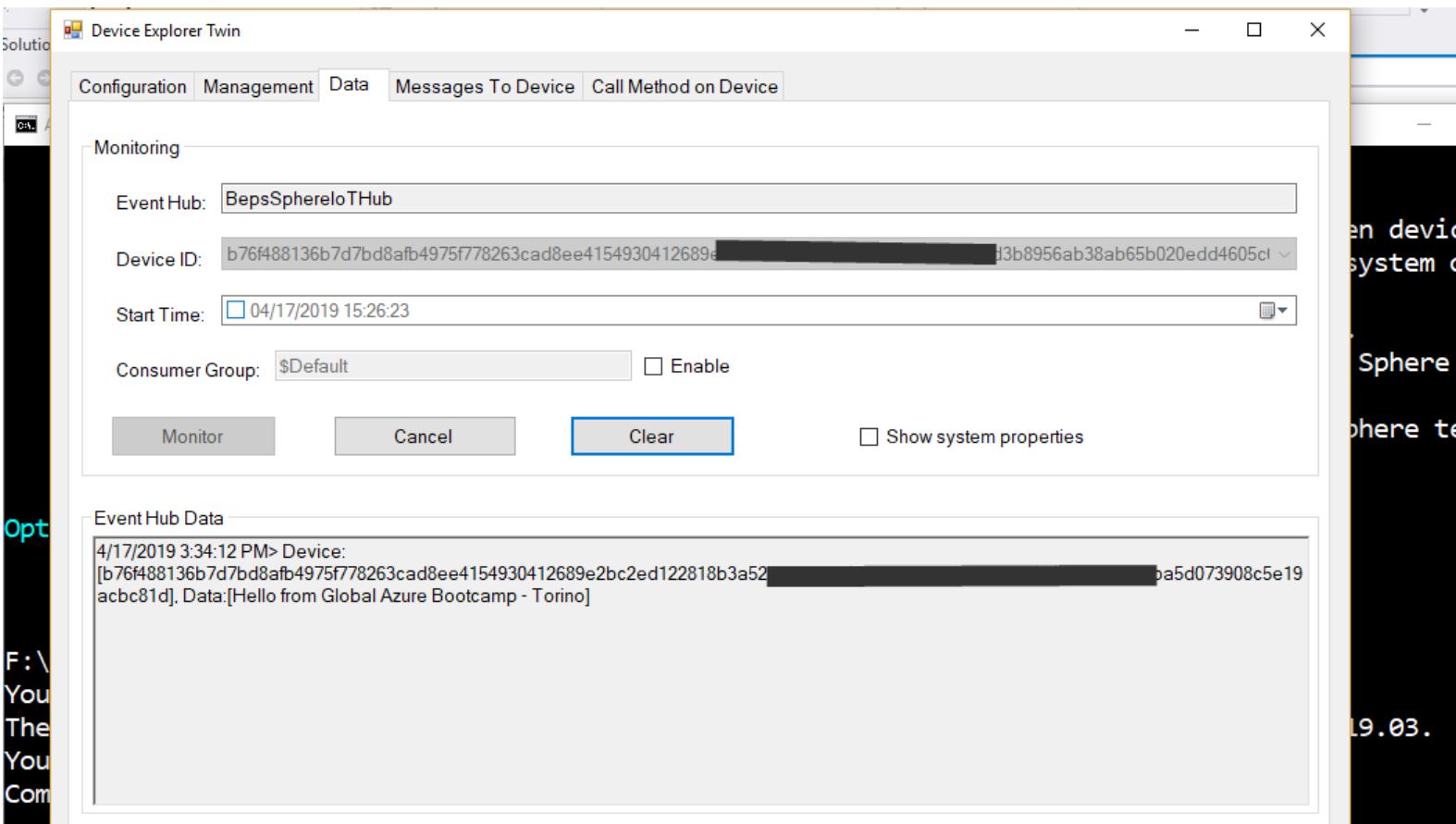
Save Discard More



Demo: IoT-Hub + Azure Sphere – Check connection



Demo: IoT-HuB + Azure Sphere – Event Hub Data



```
F:\Users\beppe\Documents>azsphere dev show-ota-config
Retrieved the over-the-air update configuration for device with ID 'B76F488136B7D7BD8AFB4975F778
8EE4154930412689E2BC2ED1' : 8190ED3B8956AB38AB65B020EDD4605C6BA5D073908C5E19ACBC81D'
--> Device group: 'System Software Only' with ID 'cd037ae5-27ca-4a13-9e3b-2a9d87f9d7bd'
--> SKU: '9d606c43-1fad-4990-b207-554a025e0869' of type 'Chip'
--> SKU: 'f57abbdcc-9340-44a9-8630-c1ee9b371ec5' of type 'Product'
Command completed successfully in 00:00:02.7560128.
```

Demo: IoT-Hub + Azure Sphere – Message to my device

The screenshot shows the Microsoft Device Explorer Twin interface. The top navigation bar includes tabs for Configuration, Management, Data, Messages To Device (which is selected), and Call Method on Device. Below the tabs, there's a section titled "Send Message to Device:" with fields for "IoT Hub" (set to "BepsSphereIoTHub"), "Device ID" (showing a long GUID), and "Message" (containing "Message to my Azure Sphere"). Two checkboxes are present: "Add Time Stamp" and "Monitor Feedback Endpoint". At the bottom, there are "Properties" and "System Properties" tabs, and a table with columns "Key" and "Value". A red arrow points to the "Message" input field. In the bottom right corner, there's a separate "Output" window showing logs from an Azure IoT client. The log entries include multiple "INFO: AzureIoT_DoPeriodicTasks calls in progress..." messages followed by one "INFO: Received message 'Message to my Azure Sphere' from IoT Hub" message. A red arrow points to this final log entry.

Device Explorer Twin

Configuration Management Data Messages To Device Call Method on Device

Send Message to Device:

IoT Hub: BepsSphereIoTHub

Device ID: b76f488136b7d7bd8afb49...bc2ed122818b3a5287d648190ed3b8956ab38ab65b020edd4605c61

Message: Message to my Azure Sphere

Add Time Stamp Monitor Feedback Endpoint

Properties System Properties

Key	Value
-----	-------

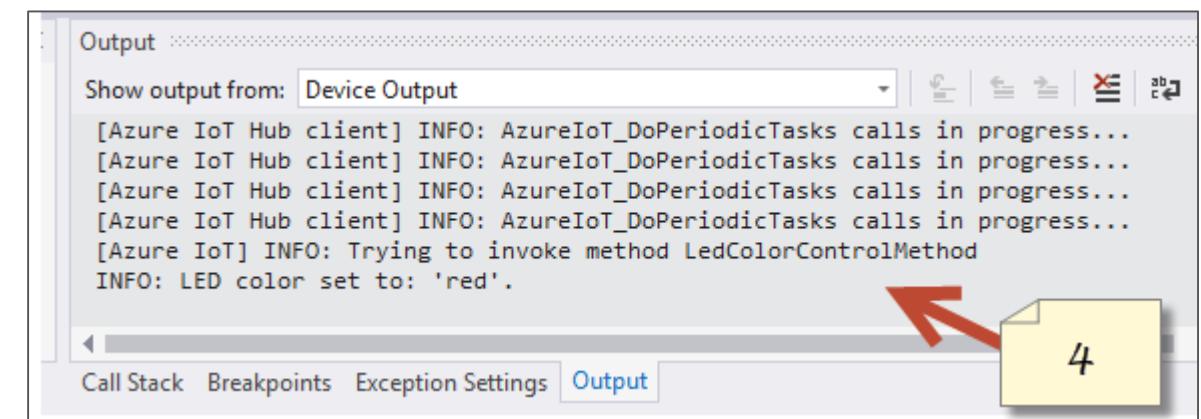
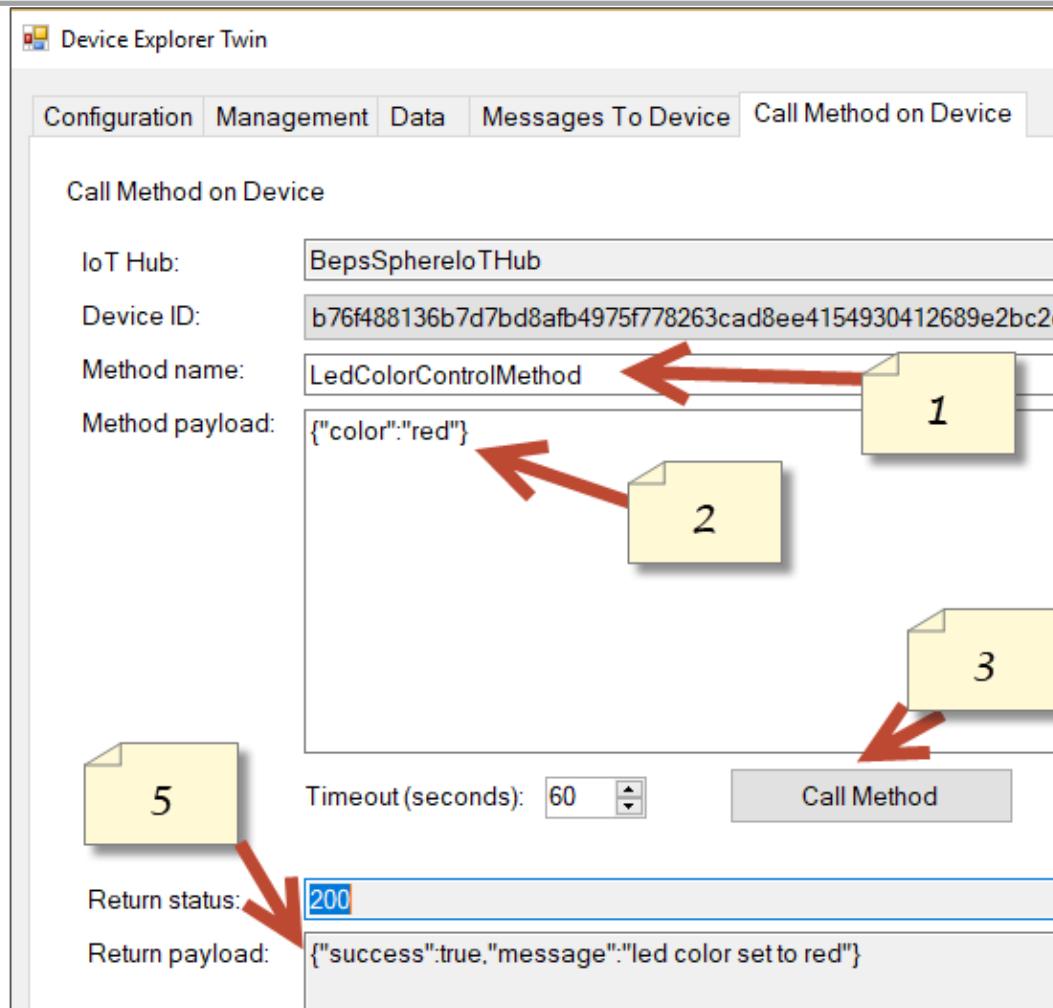
Output

Show output from: Device Output

```
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...
[Azure IoT] INFO: Received message 'Message to my Azure Sphere' from IoT Hub
```

Call Stack Breakpoints Exception Settings Output

Demo: IoT-Hub + Azure Sphere – Change method



Demo: IoT-HuB + Azure Sphere – Device Twin

The screenshot illustrates the process of updating a device twin's desired properties via the Azure Device Twin interface.

Device Twin Interface: On the left, the "Device Twin" interface shows the device ID `b76f488136b7d7bd8afb4975f77` and a dropdown showing `3b8956ab38ab65`. Below it, tabs for "Entire Twin", "Tags", "Reported Properties", and "Desired Properties" are visible. The "Desired Properties" tab is selected, displaying the following JSON:

```
{  
  "properties": {  
    "desired": {  
      "LedBlinkRateProperty": 1,  
      "WaitingTimeProperty": 120,  
      "$metadata": {  
        "$lastUpdated": "2019-04-17T13:54:29.4012994Z",  
      }  
    }  
  }  
}
```

A red box highlights the `"LedBlinkRateProperty": 1` entry. A large red arrow points from this entry to the "Send (use Json format)" button at the top right of the interface.

Output Window: On the right, the "Output" window shows the logs of the Azure IoT Hub client. The log entries are:

```
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...  
[Azure IoT] INFO: Device Twin reported properties update result: HTTP status code 204  
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...  
INFO: Received desired value 1 for LedBlinkRateProperty, setting it to 1.  
[Azure IoT] INFO: Set reported property 'LedBlinkRateProperty' to value 1.  
[Azure IoT Hub client] INFO: AzureIoT_DoPeriodicTasks calls in progress...
```

A red box highlights the line `INFO: Received desired value 1 for LedBlinkRateProperty, setting it to 1.`, and a red arrow points from this line to the corresponding entry in the "Desired Properties" JSON above.

Azure Sphere – Chi lo usa, perché



LEONI



SICUREZZA

7 proprietà rispettate

PRODUTTIVITA'

Facilità di sviluppo – VS-2017

OPPORTUNITA'

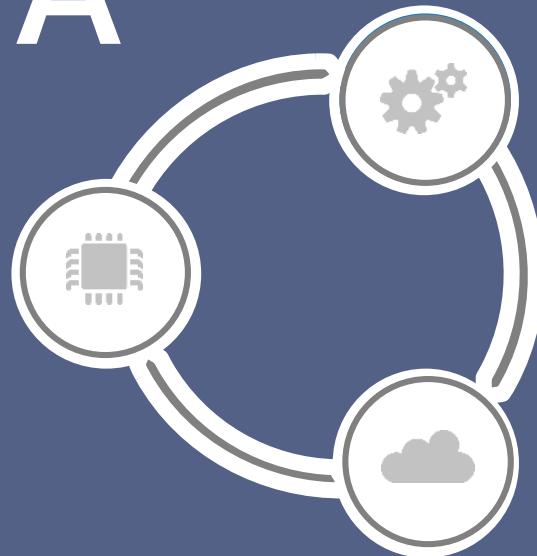
Il futuro è adesso!

Azure Sphere – Qualche riferimento

- Un tutorial dei passi che abbiamo visto dall'acquisto del dev-kit alla distribuzione via wifi.
<https://docs.microsoft.com/en-us/azure-sphere/>
- Una presentazione di come è fatta la board di sviluppo e lo "starter-kit". Oltre alla parte Hardware è interessante la rassegna di riferimenti in GitHub per integrare i vari sensori.
http://wiki.seeedstudio.com/Azure_Sphere_MT3620_Development_Kit/
http://wiki.seeedstudio.com/Grove_Starter_Kit_for_Azure_Sphere_MT3620_Development_Kit/
- Alcuni video sull'argomento:
<https://www.youtube.com/watch?v=wJgCzaiRz9w>
<https://channel9.msdn.com/Shows/Internet-of-Things-Show/Azure-Sphere-Overview>
<https://channel9.msdn.com/Shows/Internet-of-Things-Show/Azure-Sphere-Architecture-Discussion>
- BLOG di David Jones (l'MVP più attivo e prolifico sull'argomento):
<https://davidjones.sportronics.com.au/azedge/Azure-Sphere-Samples-azedge.html>

Q & A

AZURE SPHERE:
Facile da usare,
Disegnata per essere sicura!



Grazie !



Beppe Platania

CEO di BEPS Engineering S.r.l.

Twitter: @beppeplata
Blog: beppeplatania.com
eMail: beppe.platania@bepseng.it