The Hardware Security Platform Behind Azure Sphere

Doug Stiles
Sr Director, HW Engineering
Microsoft Silicon Development
Microcontrollers (MCUs)

Low-Cost single chip computers
Manufactured in fully depreciated fabs
This older low-cost technology supports:
• good compute performance
• variety of connectivity solutions
• sizeable on-chip memory

† TMS1100: 300 KHz core, 2KB ROM, 64B RAM, 23 GPIO pins
The Internet of Things and Security

MCUs are used everywhere

9 billion connected devices shipped in 2017

Estimated 30 billion connected devices by 2020

The Mirai virus was first identified in August, 2016

  Targets devices running Linux such as IP cameras, home routers, printers

  Uses these devices as bots as part of a botnet in large scale Distributed Denial of Service (DDOS) attacks

Dyn (a Domain Service Provider) was attacked in October, 2016 resulting in loss of major internet platforms and services in large parts of Europe and North America
A new Azure Sphere class of MCUs, from silicon partners, with built-in Microsoft security technology provide connectivity and a dependable hardware root of trust.

A new Azure Sphere OS secured by Microsoft for the devices 10-year lifetime to create a trustworthy platform for new IoT experiences.

The Azure Sphere Security Service guards every Azure Sphere device; it brokers trust for device-to-device and device-to-cloud communication, detects emerging threats, and renews device security.
Highly-secured connected devices require 7 properties

- **Hardware Root of Trust**: Is your device’s identity and software integrity secured by hardware?
- **Defense in Depth**: Does your device remain protected if a security mechanism is defeated?
- **Small Trusted Computing Base**: Is your device’s TCB protected from bugs in other code?
- **Dynamic Compartments**: Can your device’s security protections improve after deployment?
- **Certificate-Based Authentication**: Does your device use certificates instead of passwords for authentication?
- **Failure Reporting**: Does your device report back about failures and anomalies?
- **Renewable Security**: Does your device’s software update automatically?

![Support Levels](image)
Azure Sphere MCUs are connected, secured, crossover devices

**CONNECTED** with built-in networking

**SECURED** with built-in Microsoft silicon security technology including the Pluton Security Subsystem

**CROSSOVER** Cortex-A processing power brought to MCUs for the first time
MediaTek MT3620 – the first Azure Sphere class MCU

- 40 nm RFCMOS technology
- System-in-package (SIP)
- 164 pin DR-QFN
- 16 or 32 MB flash in package
- Single 3.3V supply
- PSU generates supply voltages for:
  - Analog
  - Fuse programming
  - Core voltage
MediaTek MT3620
The first Azure Sphere class Microcontroller

Securely isolated subsystems:

- Application Processor
- Pluton Security
- I/O peripherals
- I/O processing
- WiFi
WiFi Subsystem

Dedicated high-performance 160 MHz N9 32-bit RISC core

Dedicated OTP e-fuse block for Wi-Fi specific calibration and configuration

IEEE 802.11 a/b/g/n compliant

20MHz bandwidth in 2.4GHz and 5GHz bands

Dual-band 1Tx/1Rx mode

Built-in RX diversity support

Full TX/RX antenna diversity support
I/O Peripherals and Processor Subsystems

Two 200 MHz ARM Cortex M4 cores, each with 192kB TCM, 64kB SRAM, and integrated FPU

I/O Peripheral groups are mapped by SW to their assigned M4 core

Five “ISU” serial interface blocks configured as I2C master, I2C slave, SPI master, SPI slave, or UART

Two I2S interfaces supporting slave and TDM slave modes

Eight-channel, 12-bit, 2MS/s single-ended ADC

76 programmable GPIO (some multiplexed with other functions)

12 PWM outputs

24 external interrupt inputs
500 MHz ARM Cortex A7 with NEON and FPU support
64kB L1 instruction cache
32kB L1 data cache
256kB L2 cache
4MB system memory
Pluton Security Subsystem

200 MHz Dedicated M4 Processor

ROM for initialization and boot code

128 KB TCM for security runtime

4 Kb dedicated e-fuse for crypto keys, security state, and rollback state
Pluton Engine (Hardware Security Platform)

- RNG – Random Number Generator
- CCE – Complex Command Engine
- PKA – Public Key Accelerator
- SHA – Secure Hash Algorithm
- AES – Advanced Encryption System
- AEB – Access Enablement Block
- ECC – Elliptic Curve Cryptography

Diagram:
- RNG
- AHB-to-AXI Bridge
- AXI Bus
- Shared Memory
- CCE
- PKA
- SHA
- AES
- Fuse Control
- AEB
- Isolated ECC Key

Microsoft
Keys randomly generated and device unique
Keys in fuses and not software accessible
Crypto operations in HW
Units have HW firewalls
CPU to CPU messaging via mailboxes
Watchdog timers for failed operations
Configurations are sticky and locked
HW based attestation

Security processor is first to boot and initial code is in ROM
Application CPU has MMU
Software in separate processes
Separate CPUs and memory for Security, OS, WiFi, and I/O processing
HW error detection with SW reporting to cloud
Software is signed
No passwords
SW rollback protection

Hardware Root of Trust
Defense in Depth
Small Trusted Computing Base
Dynamic Compartments
Certificate-Based Authentication
Failure Reporting
Renewable Security
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog to Digital Converter</td>
</tr>
<tr>
<td>AHB</td>
<td>Advanced High Performance Bus</td>
</tr>
<tr>
<td>AXI</td>
<td>Advanced eXtensible Interface</td>
</tr>
<tr>
<td>FPU</td>
<td>Floating Point Unit</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input/Output</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>LDO</td>
<td>Low-voltage DropOut regulator</td>
</tr>
<tr>
<td>MCU</td>
<td>Microcontroller Unit</td>
</tr>
<tr>
<td>MMU</td>
<td>Memory Management Unit</td>
</tr>
<tr>
<td>NEON</td>
<td>ARM technology SIMD (Single Instruction Multiple Data) extension to ARM A core</td>
</tr>
<tr>
<td>OTP</td>
<td>One-Time Programmable</td>
</tr>
<tr>
<td>PSU</td>
<td>Power Supply Unit</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>ROM</td>
<td>Read Only Memory</td>
</tr>
<tr>
<td>RX</td>
<td>Receive</td>
</tr>
<tr>
<td>TCM</td>
<td>Tightly Coupled Memory</td>
</tr>
<tr>
<td>TDM</td>
<td>Time Division Multiplexed</td>
</tr>
<tr>
<td>TX</td>
<td>Transmit</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver-Transmitter</td>
</tr>
</tbody>
</table>