RISC-V Days Tokyo 2021 Autumn Demonstration

RISC-V IoT OTA and Root of Trust (RoT) Demo

Nov 17, 2021

SH Consulting Group (SHC)

Overview

This demonstration shows a full integration of secure OTA firmware upgrade and hardware root of trust (RoT) for an IoT with a 32-bit RISC-V running FreeRTOS, an open source real-time operating system. In this demonstration the RISC-V RTOS-based IoT conducts OTA via WiFi leveraging Amazon Web Services (AWS) IoT Core.

When IoT devices are deployed, the IoT service providers need to check the identity of the IoT device that is communicating with the messaging gateway. When a device is powered up for the first time, the hardware root of trust (RoT) generates key pairs for the devices, which are then used to authenticate and encrypt the traffic. The keys are generated inside the hardware root of trust (RoT) itself and are thereby protected from being retrieved by external programs.

In addition to the capabilities of a hardware root of trust and secure boot, the hardware root of trust (RoT) is also valuable just as a hardware key store. The private keys are protected by the hardware and offer far better protection than a key protected by software. The system integration leverages the PKCS#11 protocol as the interface to the hardware root of trust (RoT).

SHC demo, after generating key pairs, utilizes the FreeRTOS MQTT library and OTA agent to connect to the AWS Cloud and then performs OTA firmware update function to update new firmware to RISC-V SoC.

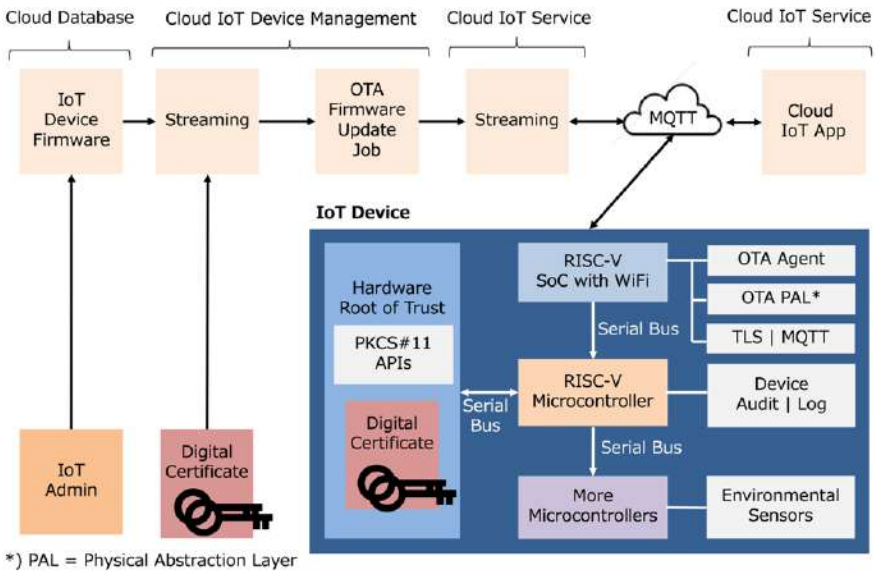

Figure 1: Full Diagram of a RISC-V AWS IoT Core Device Example

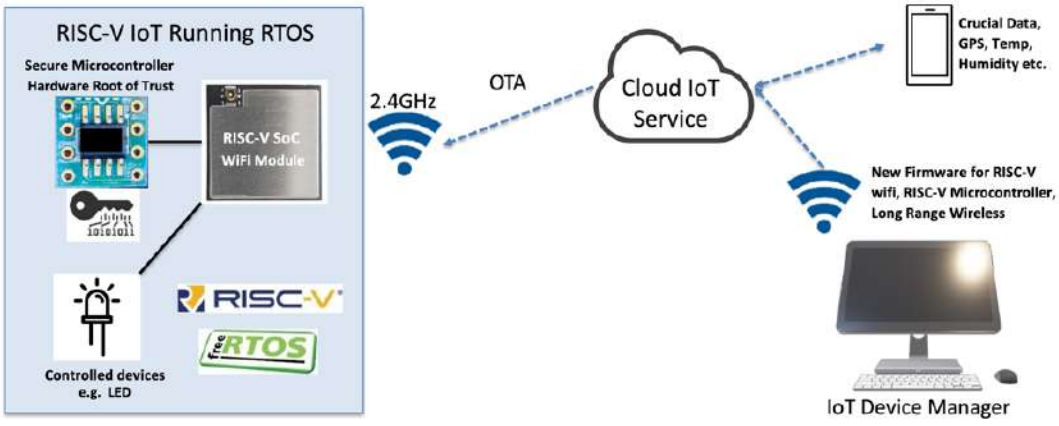

Figure 2: RISC-V OTA and RoT Demonstration Setup

Our demo system demonstrates OTA for the first RISC-V processor. A next step is to have a larger system and perform OTA for multiple processors in a secure manner.

Over-the-Air(OTA) Firmware Update

Over-the-air (OTA) firmware update allows a user to remotely update firmware to a device and redeploy the device with new firmware automatically without physical intervention. A new set of firmware is digitally signed so the system can verify the signature when it receives the file. When a programmer creates an OTA update, the OTA Update Manager service creates an AWS IoT job to notify the device that an update is available. The OTA demo application runs on the device with a FreeRTOS task that subscribes to notification topics for AWS IoT jobs and listens for update messages. When an update is available, the OTA Agent publishes requests to AWS IoT Core and receives updates using the HTTP or MQTT protocol. In SHC demo MQTT protocol is used. The OTA Agent checks the digital signature of the downloaded files, if the files are valid, installs the firmware update. After finishing firmware updating, the board is automatically reset to start new firmware.

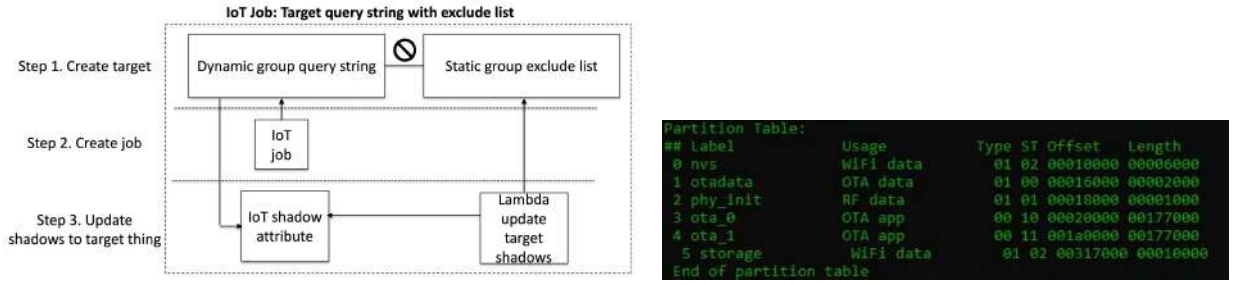

Figure 3: AWS OTA Firmware Update Process (Left) and OTA Partition Table (Right)

Measurements Made from RISC-V OTA FIrmware Update

160KHz RISC-V running QPI flash at 80MHz updated 1,135,872 bytes of RISC-V firmware image in 277 file blocks via MQTT took 2 minutes and 23 seconds or 143 seconds. This translates into 8K bytes per second. Cloud connection was made by Wi-Fi connectivity.

Memory Requirements

For secure connection to AWS IoT Core using a commercial hardware root of trust (RoT) and a RISC-V requires flash memory larger than (219.5KB + size of the application program), and random-access memory (RAM) larger than 32.87KB. For (OTA) over-the-air firmware update using hardware root of trust (RoT) and RISC-V requires flash memory larger than (251.36KB + the size of firmware), and minimum of 36.07KB of RAM. Theoretically 500KB or larger flash memory is minimum. To implement an OTA firmware update solution with confidence more than 1MB of flash memory or more is required.

Marmot = Microcontroller **A**rchitecture to **R**esist **M**alware, **O**bstructions, and **T**ampering

The following diagram shows our Marmot Wireless IoT system developed. This is an RTOS IoT system which provides: (1) Fast Response Time, (2) Low Maintenance, (3) High Reliability, (4) High Security

Figure 4: Marmot Emblem (Manufactured by Ando Cloisonne, Nagoya, Japan)

Marmot = Microcontroller Architecture to Resist Malware, Obstructions, and Tampering

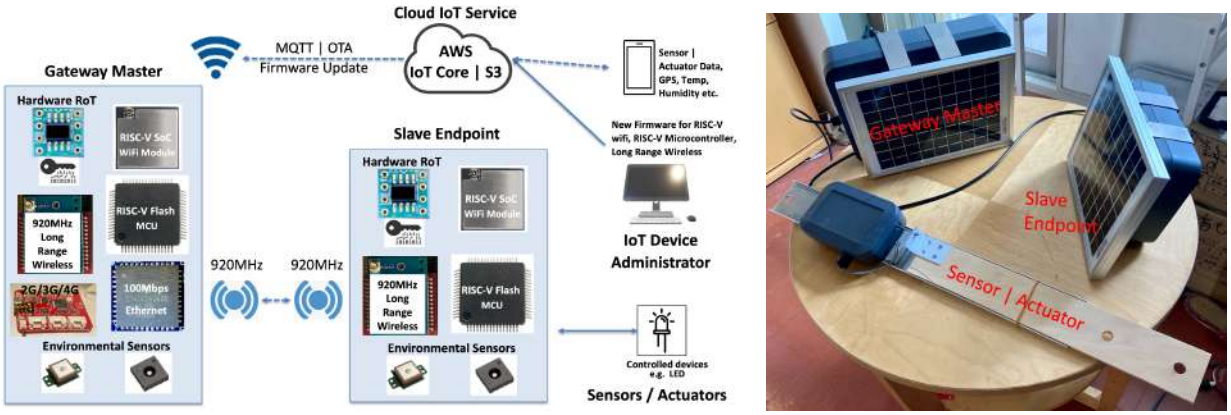

Figure 5: Marmot Wireless IoT System Example

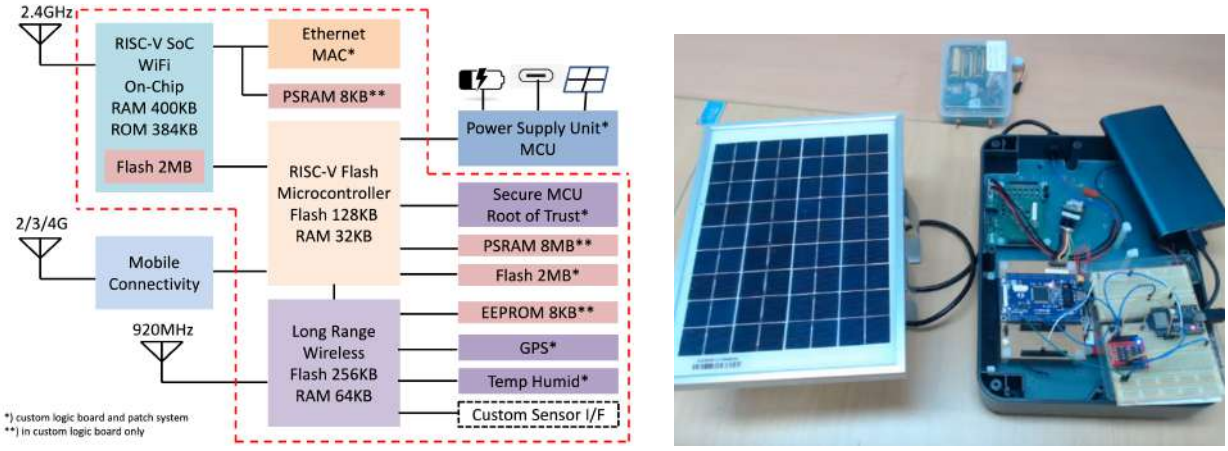

Figure 6: Marmot System Diagram and Patch Boards used for Software Development

RISC-V SoC

RISC-V SoC features Wi-Fi and Bluetooth Phy and is one of the RISC-V platforms qualified for Amazon FreeRTOS. The SoC comes with a 32-bit RISC-V Single-core processor running at 160MHz, 4MB external Flash, 384KB ROM and 400KB data SRAM, and GPIO, I2C, PWM, SPI, and UART. It has the following wireless features: (1) IEEE 11 b/g/n-compliant, (2) 1T1R mode with data rate up to 150Mbps, (3) MPDU and A-MSDU aggregation, (4) 4µs guard interval, (4) 2,412MHz to 2,484MHz operating frequency

Hardware Root of Trust (RoT)

A hardware root of trust (RoT) or a hardware root-of-trust (RoT) provides an IoT endpoint device security with countermeasures against physical tampering and side-channel attacks. The hardware root of trust (RoT) protects the private keys used by the IoT endpoint to establish the device identity, and prevents the private data from being taken out of the devices for impersonation and other malicious activities.

The particular hardware root of trust (RoT) device we used is designed to securely store multiple private keys along with their associated public keys and certificates. It supports random private key generation internally within the device to ensure that the private key can never be known outside of the device. The public key corresponding to a stored private key is always returned when the key is generated and may optionally be computed at a later time. The internal random number generator is designed to meet the requirements documented in the NIST 800-90A, 800-90B and 800-90C documents. These random numbers can be employed for usage as part of the device's crypto protocols.

Because each random number is guaranteed to be essentially unique from all numbers ever generated on this or any other device, their inclusion in the cryptographic protocol calculation ensures that replay attacks (i.e. re-transmitting a previously successful transaction) will always fail.

Cryptographic Co-Processor with Secure Hardware-based Key Storage:
– Protected Storage for up to 16 Keys, Certificates or Data
Hardware Support for Asymmetric Sign, Verify, Key Agreement:
– ECDSA: FIPS186-3 Elliptic Curve Digital Signature
– ECDH: FIPS SP800-56A Elliptic Curve Diffie-Hellman
– NIST Standard P256 Elliptic Curve Support
Hardware Support for Symmetric Algorithms:
– SHA-256 & HMAC Hash including off-chip context save/restore
– AES-128: Encrypt/Decrypt, Galois Field Multiply for GCM
Networking Key Management Support:
– Turnkey PRF/HKDF calculation for TLS 1.2 & 1.3
– Ephemeral key generation and key agreement in SRAM
– Small message encryption with keys entirely protected
Secure Boot Support:
– ECDSA code signature validation, optional stored digest/signature
– Optional communication key disablement prior to secure boot
– Encryption/Authentication for messages to prevent on-board attacks
Internal High-Quality NIST SP 800-90A/B/C Random Number Generator (RNG)
Two High-Endurance Monotonic Counters
Guaranteed Unique 72-bit Serial Number

Cloud IoT Service (Amazon IoT Core)

AWS IoT Core is a managed cloud service that lets connected devices interact with cloud applications and other devices. AWS IoT Core supports devices and messages, and processes and routes those messages to AWS endpoints and to other devices. With AWS IoT Core, applications can keep track of and communicate with all your devices, all the time, for OTA or pushing messages.

Acknowledgements

This work is based on results obtained from project, JPNP16007, commissioned by The New Energy and Industrial Technology Development Organization (NEDO). Patents and trademarks pending for Marmot System.

About SH Consulting Group

SH Consulting Group (SHC) has engineers in US, Vietnam and in Japan specialized in providing stability to RTOS, device drivers, and wireless connectivity for MCUs such as H8s, SHs, ARMs and RISC-Vs. It has been integrating OSes such as QNX, .NETMF, Linux, and Windows for MCUs and wireless solutions such as Lora, WiFi and Bluetooth for many years. They worked on Windows, Android and iOS platforms. In recent years SHC engineers enabled FreeRTOS for large semiconductor companies on ARM and RISC-V.

SH CONSULTING K.K. (JAPAN)
Tokyo Head Office: 7-18-13-502 Ginza, Chuo-ku, Tokyo, Japan 104-0061
Phone: 03-3833-3717
Tokyo Design Center: Room 202 Sunmail, 2-2038-13 Imokubo, Higashiyamato-shi, Tokyo 203-0033

SH CONSULTING VIETNAM COMPANY LTD. (VIETNAM)
(Local Name: CÔNG TY TNHH SH CONSULTING VIỆT NAM)
Quang Trung Software Park, Tan Chanh Hiep Ward, District 12 Ho Chi Minh City
Phone: 84-8-3715-0060