How to Connect to AWS IoT Core using Amazon FreeRTOS for Embedded Devices – Hands-on Workshop Using STM32L4 Discovery Kit IoT Node

Anton Shmagin (AWS)
Manuel Cantone (STMicroelectronics)
The IoT Movement

IoT is a movement where any system is able to leverage the Internet and its ecosystem

Cloud computing – Low cost embedded computers – Explosion of reliable wireless connectivity – Rapid innovation of low cost sensors

- **Nodes**
  - Smart Things (With intelligence)
  - Simple Things (Send raw data)

- **Gateways**
  - Gateway

- **Cloud**
  - Network infrastructure (Switch / Router)
  - Servers (Big data & cloud)

Steps:
- Sense
- Process
- Connect
- Power
- Secure
ST Has all the Building Blocks for the IoT

“Thing” you know how to build.

Plus what “Smart”?

Pieces from different sources and and no building instructions…

...or

IF IT’S SMART, WE'RE THERE

www.st.com
Supporting the IoT Movement

Pre-integrated SW for vertical applications
- Smart Things
- Smart Home
- Smart City
- Smart Industry

Development Ecosystem
- Code generators
- Prototyping software
- Development environments
- Debug solutions
- Simulation and analysis tools
- On-line design tools
STM32L475 Discovery Kit IoT Node
B-L475E-IOT01A

SW Libraries for STM32L4 MCU & Sensors

Low-power long-range communication (SubGHz)

Direct Wi-Fi connection to cloud servers

Environmental awareness: humidity, pressure, temp

Detection hub: motion, proximity, audio
Workshop Deliverables

• For the workshop ST will provide

**Discovery Kit IoT node**


**USB 2.0 A-Male to Micro B Cable**
STM32L4 Discovery Kit IoT Node

- STM32L4 MCU enables your IoT projects with the combination of ultra-low power and high performance

- STM32L4 Discovery Kit IoT is a compact, yet powerful board to explore various connectivity options

Next Steps

- You can add BLE, NFC and sensors features to the Amazon FreeRTOS project to create an application that can connect to the cloud and also has short range connectivity with your smartphone

- Post your projects or ideas on the ST Community website to gather feedback and get support: [https://community.st.com/community/share-your-activities/pages/overview](https://community.st.com/community/share-your-activities/pages/overview)

- Refer to Amazon FreeRTOS user guide for more examples.
Software and Other Pre-requisites
Software and Others Pre-requisites
Windows (Win7, Win8, Win10)

  - NOTE: Required for Window 7

- **Serial line monitor:** Tera Term (https://ttssh2.osdn.jp/)
  - Install from USB. \Windows\teraterm-4.99.exe
System Workbench for STM32
(requires registration to openstm32.org)

http://www.openstm32.org/Downloading+the+System+Workbench+for+STM32+installer

• install_sw4stm32_win_64bits-v2.5.exe
• install_sw4stm32_macos_64bits-v2.5.run
  • Warning: To run System Workbench for STM32 on MAC OSX systems, XCode may be required. To download it, please refer to the Apple developer website (registration as Apple Developer is required)
  • The downloaded installer is an executable binary file. Your web browser might have removed the execution right of the file. Please set the execution right to the installer file (chmod 755 install_sw4stm32.run then ./install_sw4stm32.run) OR launch it with /bin/bash (/bin/bash install_sw4stm32.run)
  • If an error message saying the installer “is damaged and can’t be opened. You should move it to the Trash.”, please modify the installation access right in the Gatekeeper. On latest version of MAC OSX, go in the terminal:

  ```
  #To disable
  sudo spctl --master-disable
  #To set the Gatekeeper access right back
  sudo spctl --master-enable
  ```

  • On older version of MAC OSX:
  • Go in the “System Preferences” > “Security & Privacy”, then select “Allow downloaded app from :” “Anywhere”. When the installation is done, restore the setting value back at “Mac App Store and identified developers”

• install_sw4stm32_linux_64bits-latest.run
  • The installer in GUI-mode requires gksudo. On Ubuntu, gksudo is in package gksu, that you can install by sudo apt-get install gksu. If gksudo is not installed, the installer can still be launched in command-line.
  • The downloaded installer is an executable binary file. Your web browser might have removed the execution right of the file. Please set the execution right to the installer file (chmod 755 install_sw4stm32.run then ./install_sw4stm32.run) OR launch it with /bin/bash (/bin/bash install_sw4stm32.run)
Tera Term Setup

- Host: [Input Field]
- Service: Telnet
  - TCP port: 22
- Serial Port: COM3
  - Protocol: UNSPEC

- SSH version: SSH2

- Setup Window
Tera Term Setup

**Serial Port Setup**
- **Port**: COM3
- **Speed**: 115200
- **Data**: 8 bit
- **Parity**: none
- **Stop bits**: 1 bit
- **Flow control**: none
- **Transmit delay**:
  - 0 msec/char
  - 0 msec/line

**Terminal Setup**
- **Terminal size**: 80 x 25
- **Receive**: LF
- **Transmit**: CR+LF
- **Terminal ID**: VT100
- **Answerback**: 
- **Coding (receive)**: UTF-8
- **Coding (transmit)**: UTF-8
- **Locale**: American
- **Code Page**: 65001

**Other Options**
- **Local echo**
- **Auto switch (VT<>TEK)**
Software and Others Pre-requisites
Mac OS: PicoCom

**Serial line monitor:** PicoCom or Screen

- Launch Spotlight by pressing Cmd + Space. Type terminal and select the Terminal app.
- In the Terminal window, enter the commands
  - `$ brew install picocom`
  - `$ ls -l /dev/tty*usbmodem*`
    - Example: /dev/tty.usbmodem413
  - `$ picocom --imap lfcrlf -b 115200 -p 1 -d 8 -c <usb device file>`
    - Example: $ picocom --imap lfcrlf -b 115200 -p 1 -d 8 -c /dev/tty.usbmodem413
Serial line monitor: PicoCom or Screen

- Launch Spotlight by pressing Cmd + Space. Type terminal and select the Terminal app.
- In the Terminal window, enter the command: `ls /dev/cu.usb*`
- In the list of devices, look for a device that contains `cu.usbserial` or `cu.usbmodem`; in the example below IoT DK is mapped to `/dev/cu.usbmodem1413`
- Launch the from the terminal the screen utility by entering the command:
  
  ```
  cesmosrv03:hack marco$ ls /dev/cu.usb*
  /dev/cu.usbmodem1413
  cesmosrv03:hack marco$ screen -L /dev/cu.usbmodem1413115200 –L
  
  cesmosrv03:hack marco$ screen -L /dev/cu.usbmodem1413
  ```

- The screen command will open a serial terminal connected to the device. Reset the board to see log messages from the device
Software and Others Pre-requisites
Linux (Ubuntu)

**Serial line monitor**: Putty or Picocom

- **Putty**
  - `$ sudo apt-get update` (This command updates the Ubuntu package list with latest one)
  - `$ sudo apt-get install -y putty`

- **Picocom**
  - `$ sudo apt-get install picocom`
  - `$ ls -l /dev/tty*usbmodem*`
    - Example: `/dev/tty.usbmodem413`
  - `$ picocom --imap lfcrlf -b 115200 -p 1 -d 8 -c <usb device file>`
    - Example: `$ picocom --imap lfcrlf -b 115200 -p 1 -d 8 -c /dev/tty.usbmodem413`
Open and Configure Serial Terminal (Putty)

- Open a Linux terminal and enter command: `dmesg`
- Open Putty
  - Note down device name for Discovery Kit
STM32CubeProg

STM32CubeProgrammer (STM32CubeProg) is an all-in-one multi-OS software tool for programming STM32 products.

Amazon Web Services
IoT on AWS
If you knew the state of everything and could reason on top of that data...

what problems would you solve?
What customers are doing with AWS IoT

- Predictive maintenance
- Wellness & health solutions
- Remote patient monitor
- Connected buildings & city systems
- Maintain device fleets
- Monitor energy efficiency
- IoT payment & connected commerce
- Safeguard manufacturing facilities
Nobody just buys IoT technology...
they seek business outcomes
Business outcomes with IoT

- New services & business models
- Products that get better with time
- Better relationship with customers
- Increased efficiency
- Intelligent decision making
- Data driven discipline

Revenue growth: IoT data drives business growth

Operational efficiency: IoT data decreases OpEx
IoT solutions are complex & multidimensional

- Devices & sensors
  - Onboard, provision, manage

- Connectivity & infrastructure
  - Connect, communicate, secure

- Analytics & insights
  - Analyze, visualize, act

- Applications & services
  - Engage, empower, delight

- Change management
  - Transform, shift culture
Our concept of IoT

Things
Sense & Act

Cloud
Storage & Compute

Intelligence
Insights & Logic → Action
Identity Service

- Bring your own Root CA and certs or let AWS IoT Core generate certificates for you
- Automatic device provisioning with Just-In-Time Registration
- Flexible and fine-grained access control with IoT policies
  - Policies can be associated with identities or registry items
  - Can control access all the way down to the MQTT topic level
Device Gateway

- Entry point into the cloud for IoT devices
- Long-lived connections for bidirectional communication
- Support for multiple protocols including MQTT, WebSockets, HTTP
- Supports SigV4, X.509 and token based authentication (via Custom Authorizors)
- Secure communications over TLS 1.2
  - Support for numerous AES and ECDHE cipher suites
Message Broker

- Scalable, low-latency, reliable message routing based on MQTT protocol
- Two-way message streaming between devices and applications
- Publish/Subscribe for decoupled devices and applications
- Support for QoS0 and QoS1 messaging
- Customizable topic space with support for wildcard topic filters
Rules Engine
Data transformation and actions

- Easy to use SQL-like language for transforming, filtering and enriching your data
- Transform—built in functions for math, string manipulation, dates, etc.
- Filter—use the WHERE clause to capture only the data you want
- Enrich—bring in context from the Device Shadow and Amazon Machine Learning or from external sources via inline Lambda execution
- Route—send your data to over 10 AWS services and third party services like Salesforce, HERE, etc.
Device Shadow

- Cloud representation of dynamic device state, e.g. temperature or RPM
- Control devices via Shadow updates like volume up or down, on/off etc.
- Devices and application notified of state change in real-time on dedicated MQTT topics (e.g., $aws/things/thing-name/shadow/update/delta)
- Query last known state for offline devices
- Automatic synchronization once devices connect
- REST APIs for applications to discover and interact with devices
- Device SDK integration for easy integration with devices
Registry

- Cloud catalog of static device meta data (e.g., Serial number, Manufacturer, etc.)

- Things that share common attributes can be associated with ThingTypes (e.g., LightBulb or Thermostat)
  - Simpler searches
  - Policies can be inherited from associated ThingTypes

- Things can be marshaled into Groups for simpler management (e.g., sensors in one building)
How can I extend AWS cloud capabilities to the edge?
AWS Greengrass extends AWS IoT onto your devices, so that they can act locally on the data they generate, while still taking advantage of the cloud.
Extend AWS IoT to the Edge

Local Messages and Triggers
Local Actions
Data and State Sync
Security
Local Resource Access
Machine Learning Inference
Protocol Adapters
Over the Air Updates

Local Message Broker
Lambda Functions
Local Device Shadows
AWS-grade security
Lambdas Interact With Peripherals
Local Execution of ML Models
Easy Integrations With Local Protocols
Easily Update Greengrass Core
How can I securely connect constrained, microcontroller-based devices?
Amazon FreeRTOS, based on the popular FreeRTOS, is a microcontroller operating system that makes small, low powered edge devices easy to program, deploy, secure, connect, and maintain.

Will it work on my chip? Does it have the functionality I need? Where do I get it? How do I start?
Amazon FreeRTOS

IoT Operating System for Microcontrollers

Amazon FreeRTOS, based on the popular FreeRTOS, is a microcontroller operating system that makes small, low powered edge devices easy to program, deploy, secure, connect, and maintain.

Will it work on my chip? Does it have the functionality I need? Where do I get it? How do I start?
Amazon FreeRTOS

IoT Microcontroller OS

Based on FreeRTOS kernel

Local Connectivity Libraries
AWS Greengrass

Cloud Connectivity Libraries
AWS IoT Core

Security Connectivity Libraries

OTA Beta & Code Signing

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
Based on #1 Real-Time Operating System for Microcontrollers

- 15 years, trusted, and widely distributed
- 40+ supported architectures
- Broad ecosystem support
- Free and open source
- Introducing version 10
- MIT Open Source License
- Improved Inter-Process Communication (IPC) capabilities with stream and message buffers
Local Connectivity Libraries
Connect with AWS Greengrass

- Local communication with edge gateways and a Wi-Fi stack, including AWS Greengrass discovery support
- Wi-Fi management library implements an abstraction layer for Wi-Fi features such as setup, configuration, provisioning, security, and power management
- Continue communicating, collecting data, and taking actions without a cloud connection
- Support for many network topologies and use cases
Cloud Connectivity Libraries

• Connectivity to AWS IoT Core
• MQTT Pub/Sub messaging
• Device Shadow support
• Take advantage of IoT Core benefits like IoT Device Management, scalable architecture, and pay as you go pricing
• Fastest way to get started on IoT microcontrollers
Security Connectivity Libraries

- Secure sockets using TLS
- Certificate-based authentication
- PKCS#11 interface for key management
- Secure by default
- No open network ports
- Only run trusted code
- Clear, modular implementation
Over-the-Air Firmware Updates

- Use AWS IoT Device Management to assign updates to groups
- Code sign new firmware images
- Stream updates to your device over MQTT
- Validate signature on device
- APIs to control installation and reboot logic
- Simple to manage groups
- Control authorship and ensure devices only run trusted code
- Memory efficient updated client
How can I manage my growing number of connected devices?
AWS IoT Device Management helps you onboard, organize, monitor, and remotely manage your growing number of connected devices.

Fast device onboarding at scale

Real-time fleet indexing and search

Monitoring and updating devices
Maintain Fleet Health

- Batch Fleet Provisioning
- Real-time Fleet Index & Search
- Fine Grained Device Logging & Monitoring
- Over the Air Updates
AWS IoT Architecture

**Things Sense & Act**
- Secure local triggers, actions, and data sync
- Gateway
- a:FreeRTOS

**Intelligence Insights & Logic → Action**
- Secure device connectivity and messaging
- AWS IoT Core
- a:FreeRTOS

**Cloud Storage & Compute**
- Fleet onboarding, management and SW updates
- Fleet audit and protection
- IoT data analytics and intelligence

- Secure device connectivity and messaging
- AWS IoT Core
- AWS IoT Device Management
- Database
- Server
- Analytics

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
How do I ensure my connected devices stay secure?
AWS IoT Device Defender is a fully managed IoT security service that enables you to secure your fleet of connected devices on an ongoing basis.

- Audit device configurations, define and monitor device behavior
- Identify drifts in security settings and detect device anomalies
- Generate alerts
- Patch security vulnerabilities
AWS IoT Device Defender

Keep Your Fleet Secure

- Audit Device Configurations
- Monitor Device Behavior
- Identify Anomalies
- Generate Alerts
- Patch Security Vulnerabilities

© 2019, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
How do I generate value from my device data?
AWS IoT Analytics is a service that processes, enriches, stores, analyzes, and visualizes IoT data for manufacturers and enterprises.

IoT data is noisy and contains gaps and false readings

Filter, process, transform, and enrich your data

Store raw data and processed data

Ad-hoc queries or sophisticated IoT analytics and visualization
AWS IoT Analytics is a service that processes, enriches, stores, analyzes, and visualizes IoT data for manufacturers and enterprises.
Amazon FreeRTOS Device Software

• NOTE: the training as given during the Technology Tour session requires temporary access to specific accounts. Generic instructions on how to get started with ST’s Discovery Kit IoT Node and Amazon FreeRTOS can be found here
  • [https://console.aws.amazon.com/iot/home#software/freertos](https://console.aws.amazon.com/iot/home#software/freertos)
  • [https://docs.aws.amazon.com/freertos/latest/userguide/getting_started_st.html](https://docs.aws.amazon.com/freertos/latest/userguide/getting_started_st.html)
B-L475E-IOT01A: AFR Support

- [https://devices.amazonaws.com/](https://devices.amazonaws.com/)
- Native WiFi Support
- OTA Updates
- Sensors Support
- BLE with EVAL-SPBTLE-1S (contact your ST representative for availability)
- Code currently hosted on Teamspace under my.st.com (account creation required)
Releasing Your Creativity

www.st.com/stm32