Building low-power IOT devices using standard microcontrollers

Objectives

- Introduce the IoT ecosystem
- Learn how to deploy an local open source IoT Platform
- Describe the most used IoT Edge to Cloud Protocols (MQTT, MQTT-SN and CoAP)
- Explore particularly heinous IoT focused attacks and security provisions at each level of stack (physical devices, communication systems and networks)
- Learn how to configure the LwIP (with MQTT), FreeRTOS and MbedTLS for an STM32 IoT application
- Understand the architecture of the Amazon FreeRTOS IOT libraries

Labs will be conducted on STM32-based boards connected through WiFi or Ethernet to a private cloud server

Prerequisites

- Familiarity with C concepts and programming targeting the embedded world
- Basic knowledge of embedded processors
- Basic knowledge of multi-task scheduling
- Basic Concepts of Cryptography
- Basic knowledge of STM32 microcontrollers

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
    - One Online Linux PC per trainee for the practical activities.
    - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
    - QEMU Emulated board or physical board connected to the online PC (depending on the course).
    - Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
    - One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - An installation and test manual is provided to allow preinstallation of the needed software.
    - The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

First Day

Introduction to IoT

- IoT potential
- IoT Architecture and Core IoT Modules
- Functional blocks of an IoT solution
- The Essentials for Building IoT platform
- Cloud Providers

Exercise: *Install and configure an open-source IoT platform*

LwIP introduction

- Overview
- Buffer and memory management
- LwIP configuration options
- Network interfaces
- MAC and IP address settings
- IP processing
- UDP processing
- TCP processing
- Interfacing the stack
- Application Program Interface (API)
- Standalone
- Netconn and BSD socket library

MQTT Protocol

- Publish-subscribe
- Architecture details
- Packet structure
- Communication formats

Exercise: *Connect and publish CPU temperature*
Exercise: *GPIO control over MQTT*

Second Day

MQTT-SN

- Architecture and topology

- Transparent and aggregating gateways
- Gateway advertisement and discovery
- Differences between MQTT and MQTT-SN

Constrained Application Protocol

- CoAP architecture details
- CoAP Messaging Formats

IoT Security

- IoT cyber attacks
- Physical and hardware security
  - Key management and trusted platform modules
  - Processor and memory space
  - Storage security
  - Physical security
- Cryptography
  - Symmetric cryptography
  - Asymmetric cryptography
  - Cryptographic hash (authentication and signing)
  - Public Key Infrastructure
  - Network Stack “ Transport Layer Security
- Best practices

MbedTLS Introduction

- Encryption/Decryption module
- Hashing Module
- RNG module
- SSL / TLS communication module
- TCP / IP communication module
- X.509 module

Exercise: *Two-way SSL connection using TLS with MbedTLS*

Third Day

Amazon FreeRTOS

- Amazon FreeRTOS Architecture
- FreeRTOS Kernel Fundamentals Overview
- Amazon FreeRTOS Libraries
- Amazon FreeRTOS Console

Amazon FreeRTOS Libraries

- Porting Libraries
- Application Libraries
- Common Libraries
  - Atomic Operations
  - Linear Containers
  - Logging
  - Static Memory
  - Task Pool
- Configuring the Amazon FreeRTOS Libraries
- Bluetooth Low Energy
- AWS IoT Device Defender
- AWS IoT Device Shadow
- AWS IoT Greengrass

- MQTT (v2.0.0 and v1.0.0)
- HTTPS
- Over-The-Air (OTA) Agent
- Public Key Cryptography Standard (PKCS) #11
- Secure Sockets
- Transport Layer Security (TLS)
- Wi-Fi

Amazon FreeRTOS Porting

- Porting FreeRTOS Kernel
- Wi-Fi
- TCP/IP
- Secure Sockets
- PKCS #11
- TLS
- MQTT
- HTTPS
- OTA
- BLE

Renseignements pratiques

Inquiry : 3 days