Bluetooth Low Energy, LoRa, 868, Radio Protocols, Arm Cortex Devices, 16-bit RISC MCUs, debugger, Android, Kotlin, Python, Embedded C, RTOS.

If any of these keywords resonate with you, then we have a good fit!

Argus Security (https://www.argussecurity.it/about-us/) is a Halma company part of the Orama brand that focuses on wired and wireless fire detection alarms. Located in Muggia in via del Canneto 14, the company internally develops hardware and firmware for the embedded devices and the interface software to monitor and configure the whole system.

Note: all the activities will be carried out both on-site and in remote.

Path 1

Embedded Firmware Engineering

The goal of the activity is to evaluate the performance of radio SOCs/SIPS in current Argus systems.

Step1: Configuring SOC/SIP ARM Boards to allow interoperation with Argus Radio System. The goal of the activity is to be able to receive and send frames according to a simplified Taurus radio protocol and printing debug data over serial port.

Step2: implementation of whitening algorithm and performance evaluation in real time.

Step3: implementation of clock drift compensation algorithm and evaluation in real system.

Notes: Initially IDE managed project, the goal is to move toward makefile configuration for CI/CD pipelines.

Driver: Argus is evaluating and selecting new technologies for the development of next generation of fire protection systems and the performance evaluation of radio COTS is part of the process. CI/CD pipelines short the time to market for incremental features.

Keywords: ARM, CortexM0+, Cortex M4, RTOS, ISM, SEGGER, ST-link, Atlassian, Git, narrowband, embedded C, Assembler.

Path 2

Software Defined Radio

The goal of the activity is to build radio scanners for Argus Radio system using SDR dongles.

Step1: Select a toolchain to perform SDR (Gnu radio, MATLAB, C++, …). Build a receiver on a single channel for a simplified version of Taurus radio protocol. Decode signals according to the simplified protocol.

Step2: Decode signals on a full version of Taurus radio protocol.

Step3: Decode all the channels of Taurus radio protocol.

Notes: the SDR activity will consist only in sniffing the radio protocol, not in the transmission of data.

Driver: SDR is a form-effective alternative to full custom radio scanners and can be used **both** in the field for customer support or site surveys **and** in the R&D flow to analyse and debug new products.

Keywords: C++, python, Matlab, SDR, subGHz, ISM, GFSK, whitening, pseudo-random sequencies, data rate, sample rate, constellation, …

Path 3

BLE 5.2 systems (1)

The goal of the activity is to build a simple BLE radio system and evaluate performances.

Step1: Define a basic profile (GATTs etc) for a smoke detection system

Step2: implement the profile on a dev board

Step3: set up a radio network scanner to debug the GATT and the dev board

Step4: evaluate the performance of a basic BLE system (few peripheral and a central node)

Driver: Argus is evaluating and selecting new technologies for the development of next generation of fire protection systems and Bluetooth Low Energy is a radio standard that is wide-word accepted and provides interoperability with many consumer devices. It can be used both for providing a wireless user experience to the end user and to setup the backbone of new radio systems.

Keywords: C, BLE, GATT, ATT, profiles, SEGGER, JLINK, serial port, low power, RTOS.

BLE 5.2 systems (2)

The goal of the activity is to build a simple BLE mesh radio system and evaluate performances.

Step1: Define a basic mesh profile for a smoke detection system

Step2: implement the profile on a dev board

Step3: set up a radio network scanner to debug network

Step4: evaluate the performance of the mesh

Driver: Argus is evaluating and selecting new topologies for the development of next generation of fire protection systems and Bluetooth Low Energy provides native mesh support. It can be used to provide self-healing and self-configuring networks and is world-wide accepted.

Keywords: C, BLE, GATT, ATT, profiles, MESH, ISM, SEGGER, JLINK, serial port, low power, RTOS.

Path 4

Interfacing Android with BLE

The goal of the activity is to write a basic app to interface with some BLE dongle using Android Studio and Jetpack Compose. Also, MAD framework will be enforced.

Step1: Set up a minimum interface to perform basic tasks

Step2: Interface with a third party Bluetooth LE device (STM BLE dev boards)

Step3: Interface with ARGUS Bluetooth LE device

Step4: Display data over User Interface

Step5: App performance profiling over emulate and real devices.

Driver: Argus aims to ease the installation and survey process for the installer providing smart tools to inspect sites. Such tools rely on Bluetooth devices and mobile apps to provide user interfaces. Moving from legacy apps to modern android development is a key factor in keeping market's pace and provide the best User Experience.

Keywords: BLE, Android, JetpackCompose, Kotlin, Coroutine, async tasks, AndroidStudio.