Embedded System Labs (嵌入式系統實驗)　(EE3021)

Meeting:　MD 303（Thursday 2:20-5:10）　　Lab: MD 305

Instructor: Prof. Sheng-De Wang, sdwang@ntu.edu.tw

TA: 劉人廣（f08921039@ntu.edu.tw ）　　Courses web: https://cool.ntu.edu.tw/

[Course Objective]

This course is designed for Junior and Senior of undergraduate students. It will acquaint students with the working principles of embedded systems with an RTOS, Linux operating systems, and embedded system programming with C/C++ Language. The outcomes of the course will let students know how to integrate embedded hardware, software, and middleware (such as OS libraries) to meet the functional requirements of embedded applications.

[Course Content]

The course will start with an introduction to embedded systems, processors, input/output systems, hardware/software tools, concepts of operating systems. Laboratories are provided to guide students to be familiar with programming and debugging tools for embedded systems. STM32L4 IoT node and *Raspberry Pi 3/4* (also optional Nvidia Nano), will be used as development boards.

1. ARM Architecture Overview
2. Introduction to CMSIS-RTOS2, freeRTOS and their API.
   STM32CubeIDE Introduction
   Lab: STM32CubeIDE and Device Configuration, LED blinky and Synchronization
3. STM32 Iot Node Onboard Sensors, BSP
   Serial Communication: UART, SPI and I2C
   Socket Programming TCP/IP Labs
   Lab: IoT node networking - wifi
4. Development Environment for Embedded System using RPi
   BLE Introduction
   Lab: BLE programming – BLE Central
5. BLE programming – BLE peripheral (BLE GATT server)
6. STM32 timers and PWM, Tools: Logic Analyzer
   Lab: STM32 PWM and Logic Analyzer
7. Data Acquisition, DMA, Interrupt Handling
8. CMSIS DSP programming
   Term Project Proposal
9. ARM Assembly Programming
10. Introduction to Raspberry Pi/Linux GPIO
    IoT device connection via one-wire Serial Interface
    Linux Device Driver Example: IIO Kernel Drivers
    Interrupt Driven IO and Polling IO
11. Final Projects (Development Boards: STM32L4 IoT (required), RPi and Nvidia Nano (optional)
    Topic: IoT Data Processing and Applications

[Grading]

Homeworks + Labs 33%, Term projects 33%, Quiz/Exam 33%, Course Participation (Including Discussions and Equipment Maintenance) 1%

[Prerequisites]

Computer programming (C/C++), Computer Architecture or Operating Systems (recommended, not mandatory)

Development Boards

1. STM32L4 For IoT node
   CPU Architecture: Cortex-M
   Reference URL: https://www.st.com/en/evaluation-tools/b-l475e-iot01a.html
2. Raspberry Pi 3
   CPU Architecture: Cortex-A
   Reference URL: https://en.wikipedia.org/wiki/Raspberry_Pi
3. Nvidia Nano
   CPU Architecture: Cortex-A
   Reference URL: https://www.nvidia.com/zh-tw/autonomous-machines/embedded-systems/jetson-nano/

Hardware Architecture of ARM (https://en.wikipedia.org/wiki/ARM_architecture)

- 32-bit architectures (legacy)

Version: ARMv6, ARMv5, ARMv4T, ARMv3, ARMv2

- 32-bit architectures (Cortex)

Version: ARMv8-A, ARMv8-M, ARMv7-A, ARMv7-R, ARMv7E-M, ARMv7-M, ARMv6-M

- 64/32-bit architecture

Introduced 2011

Version: ARMv8-A, ARMv8.1-A, ARMv8.2-A, ARMv8.3-A

Note:

Architecture -> Core -> microarchitecture -> System On Chip

For example,

- RPi 2/RPi 3 use Broadcom BCM2837 SOC.

ARMv8-A -> core Cortex-A53 -> Design by Broadcom -> Broadcom BCM2837 (1.2G/1.4G Hz 64-bit quad-core ARM Cortex-A53 processor)

- RPi 4 uses Broadcom BCM2711 SOC.

ARMv8-A -> core Cortex-A72 -> Design by Broadcom -> Broadcom BCM2711 (1.5G Hz 64-bit quad-core ARM Cortex-A72 processor)

Term Project:

Topic: Data Processing and Applications of Internet of Things

Development Boards:

- Required: STM32L4 For IoT node
- Optional: RPi, Nvidia Nano
- Required: at least one wireless communication protocol should be used: BLE/Wifi/Sub-GHz RF