CÔNG TY TNHH DEVLINUX
ĐÀO TẠO LẬP TRÌNH EMBEDDED CHUYÊN SÂU
Địa chỉ: LK05B No10 Khu Đồng Dưa, Quận Hà Đông,TP Hà Nội
ĐT: **0877123348** mail: **devlinux.edu@gmail.com**

BootLoader Mock Project

1. Assignment:

Using S32 Design Studio to write a boot loader program to upgrade the firmware through UART.

2. Explanation:

Bootloader (BOOT APP) is a small application which allows user to load new application (USER APP) from PC via UART to FLASH, and finally jumps to this new application to run it.

- BOOT APP is located at the top (BOOT area) of FLASH.
- USER APP is located behind BOOT area of FLASH(APP area).
- When starting, the program checks whether is the boot button pressed or not to run BOOT APP or USER APP.
- When the BOOT APP is running:
    - It sends a request to PC through UART for asking USER APP (srec file) to be sent.
    - User uses terminal to send USER APP to the board through UART.
    - BOOT APP receives the USER APP and locates it on APP area.
    - After the allocation is complete, the BOOT APP will send information to PC for asking user to reset board without boot button pressing to run new USER APP.

3. Plan:

- Study Srec format, write code to parser Srec-file (2 days)
- Study UART1 peripheral of S32K144, write a program to receive Srec-file sent from PC (2 days)

- The UART1 configuration is: Baud rate: 115200, Data bit: 8; Parity: None; Stop bits: 1; Flow Control: None.
- Support Queue to receive data via UART (2days)
- Using Srec parser code to parser Srec-file sent from PC, then flash data to APP area (1 day). The source code related to Flash module will be provided.
- Accomplish the assignment. (2 days)

4. Requirements:

- Separate to APP + BOOT, UART, QUEUE, FLASH, SREC layers (UTILITY if needed)
- Support Queue to receive data via UART(each queue elements will be used to store a SREC line, maximum number of queue elements is 4)
- Not use sharing global variables between layers
- Violating any reqs will be marked as 0