

## Název laboratoře

Základy programování a ladění aplikací pro Kinetis MKL05 v prostředí KDS, komunikace přes UART

## Cíl laboratoře

Hlavním cílem laboratoře je seznámit se i) s laboratorním přípravkem obsahujícím mikrokontrolér Kinetis řady MKL05 a ii) se základy tvorby a ladění aplikací pro MCU Kinetis v prostředí Kinetis Design Studio (KDS).

## Pracovní postup

1. Spustíte KDS (Kinetis Design Studio 3.0.0 IDE).
2. V nabídce "File/New" zvolíte "Kinetis Project", zadejete název svého projektu (např. Lab1) a zvolíte "Next".
3. V nabídce "Devices" zvolíte typ mikrokontroléru, se kterým budete pracovat. (Processors->Kinetis L->MKL0x->KL05Z (48MHz)->MKL05Z32xxx4). Klikněte na "Next", na další kartě nastavte volbu Kinetis SDK na "None". Poté již můžete zvolit "Finish" - vytvoří se nový projekt.
4. Vlevo vidíte okno "Project explorer" s Vámi vytvořeným projektem. Rozbalte strom projektu (klikem na malou šipku nebo dvojklikem na název). Nás zajímá větev "Sources", v níž je prozatím jen soubor "main.c", tj. soubor obsahující zdrojový text programu budoucí (vestavné) aplikace. Nahradíte obsah implicitního souboru "main.c" obsahem main.c se zadáním dané laboratorní úlohy (realizovat přes clipboard nebo uložit do souboru na disk a následně vložit do projektu).
5. Doplníte "jmeno" a "login" v hlavičce main.c "Reseni vytvoril a odevzdava: (jmeno, login)".
6. Doplníte zdrojový text v souboru "main.c" o části umožňující chod aplikace s níže specifikovaným chováním:
  - o Prostřednictvím terminálu běžícím na PC a připojeném přes sériovou linku k rozhraní UART na mikrokontroléru (MCU) bude uživatel aplikace zadávat z klávesnice PC písmena a číslice.
  - o Tyto znaky, které budou na straně MCU přijímány modulem přijímače UART, bude aplikace přes vysílací modul UART zasílat zpět na terminál (aby se po zadávání zobrazily v terminálu). Aplikace bude příchozí znaky ukládat do pole "login", a to v pořadí jejich příchodu.
  - o Po zaplnění pole (tj. po přijetí 8 znaků zadaných z klávesnice PC) aplikace porovná pole "login" s obsahem referenčního řetězce (ten je v programu definován jako pole 8 znaků "corrl" - to inicializujete textovým řetězcem s obsahem Vašeho loginu).
  - o Při shodě řetězců, tj. shodě řetězce zadaného v terminálu s referenčním řetězcem, aplikace "pípne", např. pro signalizaci povolení přístupu. K "pípnutí" lze využít volání funkce beep().
  - o V případě neshody řetězců aplikace např. "STOP" i) vypíše, pomocí vysílače UART, na terminál PC vhodný text, např. "STOP" informující o zamítnutí přístupu a ii) umožní opakované zadání znaků.
7. Postupujte podle komentářů v souboru "main.c" na příslušných místech.
8. Funkci pro příjem znaků implementujte pomocí dotazovací smyčky (angl. polling loop).
9. Program můžete přeložit/sestavit pomocí "Project/Build All" z menu či stiskem <Ctrl+B>.
10. Po úspěšném překladu spustíte debugger: "Run/Debug" z menu či <F11>.
11. Nyní je systém připraven pro běh či ladění programu. Můžete program spustit a nechat běžet ("Run/Resume"), trasovat a sledovat reakce (<F5>, <F6>) atd. V případě zjištěné nefunkčnosti Vašeho řešení využijte k odhalení chyb ve Vašem zdrojovém kódu ladicích prostředků poskytovaných prostředím KDS.

## Nastavení komunikace mezi USB@PC a UART@MCU:

Jako terminál pro komunikaci mezi PC a MCU přes rozhraní UART můžete použít např. aplikaci putty.

Komunikační parametry nastavte takto:

- 115200 Bd,
- 8 bitů,
- 1 stop bit,
- žádná parita,
- připojení na COMx (konkrétní hodnota "x" se může lišit pro jednotlivá PC - prosím zjistěte),
- řízení toku: žádné.

## Podpůrné materiály:

- 4. přednáška: Sériová asynchronní komunikace pomocí UART,
- Dokumentace k rozhraní UART (viz KL05 Sub-Family Reference Manual od str. 635),