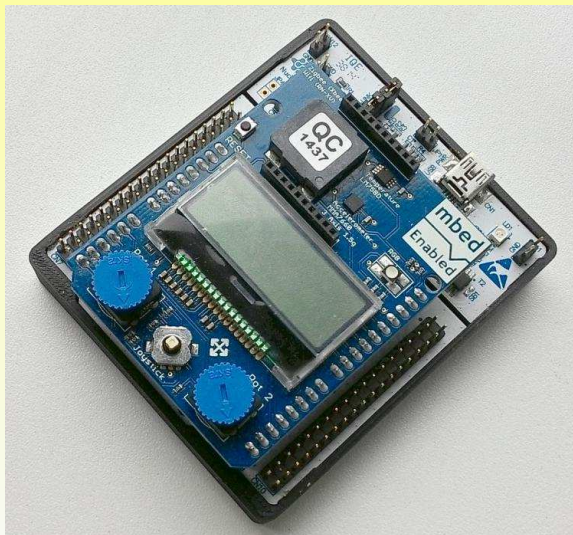


KAE/MINA - cvičení v 2017/18

Ing. Petr Weissar, Ph.D.

Ing. Petr Krist, Ph.D.



Plán cvičení – aktivity v laboratoři

1. Úvod, rozdělení kitů, opakování samostatně
 - GPIO - blikání LED, tlačítko,
 - SysTick,
 - USART
2. Pokročilé periférie ARM Cortex M
3. FPU - nastavení kompilátoru, porovnání rychlosti
4. Přerušení advanced - priority, blokování a řešení problémů
5. Problém atomických operací, řešení pomocí bit-banding
6. DMA - přenos bloku paměti, USART
7. DMA 2 - využití A/D převodníku a bufferu dat
8. Problematika Low-Power režimů
9. Privilegovaný a neprivilegovaný režim
10. SVC
11. Pokročilé techniky programování – zásobníky, semafore, instrukce LDREX/STREX, paměťové bariéry ...
12. RTOS - praktické řešení
13. Rezerva

Plán domácí přípravy

- I. Navázání znalosti z KAE/MPP
 - Spuštění známého prostředí Atollic TrueStudio (IDE+kompilátor+debugger) - Windows (příp. Linux)
 - Zopakování základní bloků práce s mikrokontrolérem
 - Jednoduchá práce s GPIO - blikání LED, čtení tlačítek.
 - Časování pomocí SysTick na 1ms
 - Sériová komunikace s PC pomocí USART a (USB virtuálního) COM portu
- II. Procvičení známých částí z KAE/MPP
 - Timer, PWM, řízení jasu LED, generování tónu
 - Použití LCD – připojen na SPI, má vlastní řadič
- III. Vyzkoušení efektu použití FPU pro float výpočty – výpočet Mandelbrotovy množiny
- IV. Ověření funkce bit-bandingu
- V. Ověření funkce priorit přerušení
- VI. Ověření činnosti DMA
- VII. Ověření spolupráce DMA a A/D převodníku
- VIII. Rezerva - doplnění chybějících věcí ze cvičení/příprav
- IX. Ověření funkce SVC a dalších pokročilých technik
- X. Prostor pro samostatnou práci - TODO
- XI. Prostor pro samostatnou práci - TODO
- XII. Prostor pro samostatnou práci - TODO

Organizace výuky, práce v laboratoři

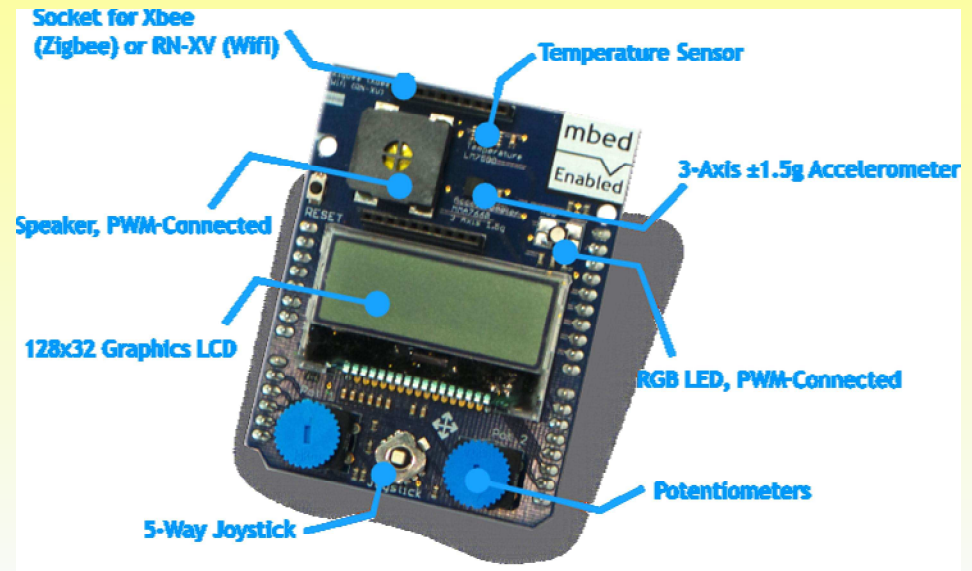
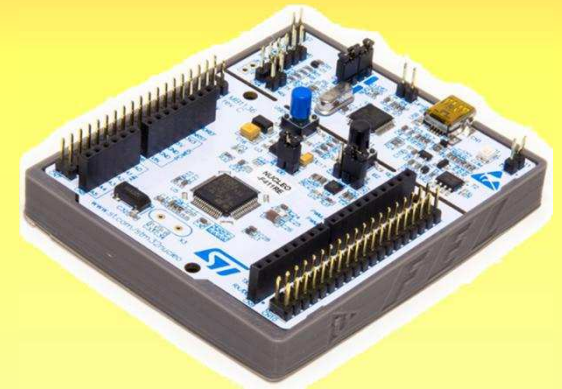
- V laboratoři pracujeme samostatně/ve dvojicích
 - Společný sdílený síťový adresář – disk T:
 - Domácí adresář studentů H:
 - Přilogování pomocí Orion-přístupu
 - Další síťové disky (R/O) - Z: (SW, podklady), X: (bat)
- HW společný – zapůjčen studentům
 - Předpokládá se část práce (příprava) doma
 - Uzavřena „smlouva o výpůjčce“
- Možno vlastní projekt založený na ARM Cortex-Mx procesoru
 - Nutno na počátku semestru schválit vyučujícím
 - Rozsahem musí odpovídat minimálně úkolům/látce ze cvičení
- Bezpečnost práce
 - Studenti musí mít 50-tku v laboratořích
- Bezpečnost elektroniky a zařízení
 - Kromě dodaných kitů veškeré další připojování HW až po **souhlasu cvičícího**

Podmínky zápočtu

- Splnění úkolů na cvičení
 - Možno (někdy nutno) dokončit v rámci domácí přípravy
- Znalost probírané problematiky na cvičeních i z "domácí přípravy"
 - Splněné kroky domácí přípravy
 - Na konci slajdů „domácí příprava“ seznam vyzkoušeného
 - Na počátku slajdů „cvičení“ seznam, co se má umět
 - Nesplnění jednotlivých etap přípravy ovlivní známku ze cvičení, se kterou se pak „jde“ ke zkoušce
- Vrácený HW v pořádku
- U vlastního HW vhodné konzultace o samostatné práci
 - Doporučeno po 3-4 týdnech

Administrativa zapůjčení kitů

- Připraven formulář o "Zápůjčce"
 - Jedna kopie zůstává, druhou má student
- Kit obsahuje v krabičce
 - Nucleo STM32F411RE
 - MBED Application shield
 - Mini USB kabel
- Vracení HW na konci semestru
 - Ideálně ve 13. týdnu po předvedení funkčních úkolů/cvičení



Zdroje informací

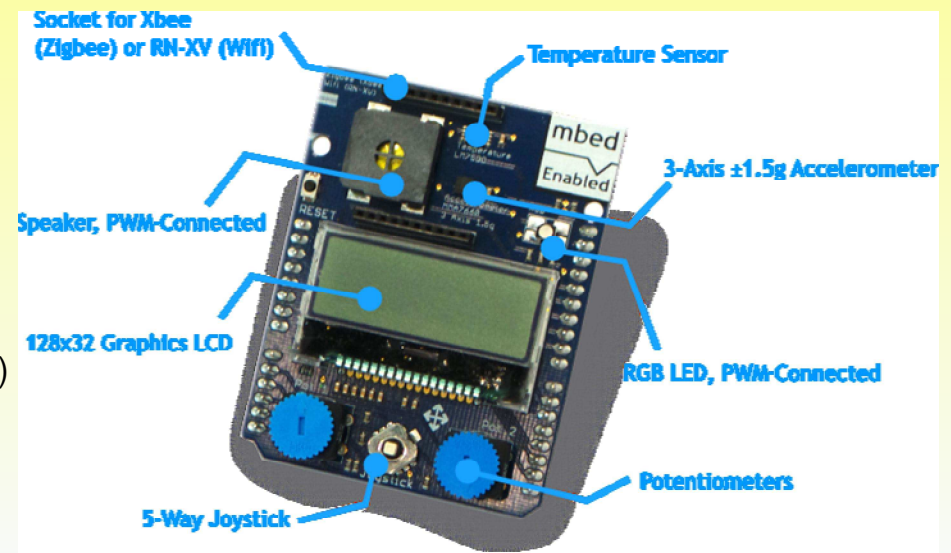
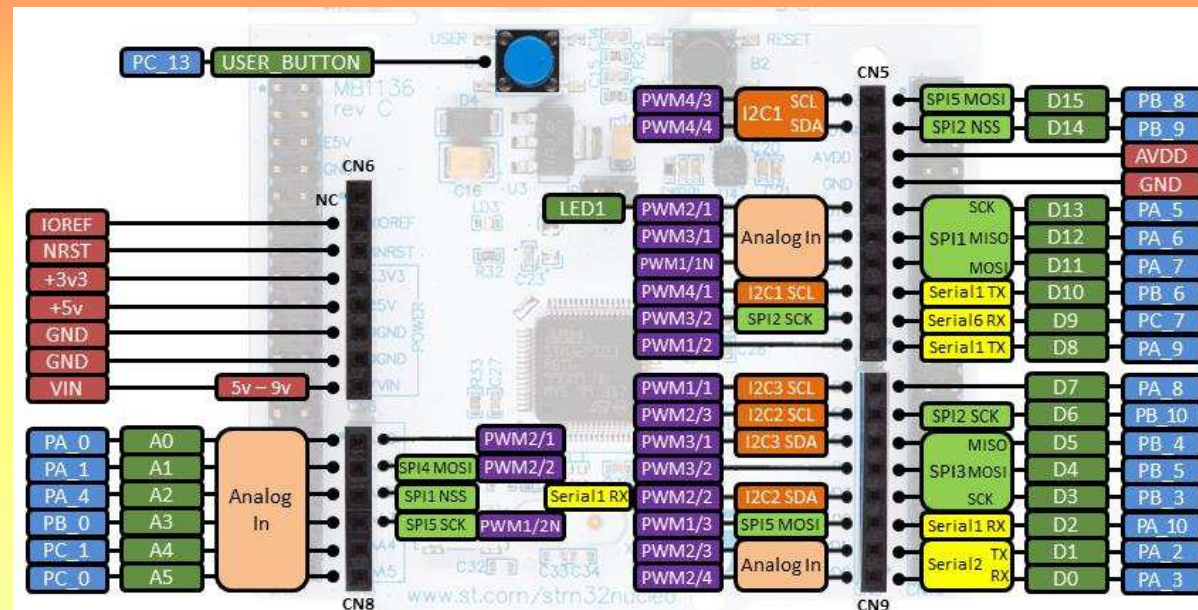
- Dostupné v elektronické podobě – typicky PDF:
 - Internet – www.st.com, mbed.org, ...
 - Vybrané na CourseWare nebo zde na Z:\podklady\MPP\...
- Podklady procesor:
 - Reference manual – RM0383 - STM32F411xC_E advanced ARM - V1_0 - en.DM00119316.pdf
 - Datasheet - DS10314 - STM32F411xC STM32F411xE - v6_0 en.DM00115249.pdf
 - Společný pro STM32F411xC STM32F411xE
 - ARM® Cortex®-M4 32b MCU+FPU, 125 DMIPS, 512KB Flash, 128KB RAM, USB OTG FS, 11 TIMs, 1 ADC, 13 comm. interfaces
 - Programming manual - PM0214 - F3 F4 L4 Programming manual - V5_0 en.DM00046982.pdf
 - STM32F3, STM32F4 and STM32L4 Series Cortex®-M4 programming manual
- Podklady Nucleo modul (<https://developer.mbed.org/platforms/ST-Nucleo-F411RE/>)
 - UM1724 - STM32 Nucleo-64 board - V11_0 en.DM00105823.pdf
 - Schematic - Nucleo schematic - rev C - MB1136.pdf
 - Pozor, Nucleo desky jsou rev. C-2 nebo C-3
- Podklady pro MBED Shield:
 - Schematic - NSHLD_SCH_1600xx_A.0_Basic.PDF
- Komponenty
 - Akcelerometr - MMA7660FC.pdf
 - <http://www.nxp.com/products/sensors/accelerometers/3-axis-accelerometers/1.5g-low-g-digital-accelerometer:MMA7660FC>

Opakování z KAE/MPP

- Pro psaní aplikací využíváme Atollic TrueStudio Lite
 - V laboratoři v. 8.0.0, příklady funkční od 7.1.x
 - Stáhnout buď <https://atollic.com/truestudio/> nebo DropBox
 - https://www.dropbox.com/s/48bqyrgml8m2wm9/Atollic_TrueSTUDIO_for_ARM_windows_x86_v8.0.0_20170621-1519.exe?dl=0
- Debug pomocí vestavěného ST-Link modulu na Nucleo desce
- Využití UART procesoru
 - Rx a Tx signály připojeny na ST-Link, který komunikuje s ovladačem virtuálního COM portu v PC
 - Na straně PC použít terminál Putty nebo Hercules
- Blikání on-board LED
 - Připojena na PA5
 - Časování pomocí for cyklu
 - Časování pomocí SysTick – nastavit na přerušení 1ms a inkrementovat "ticks"

MBED Shield

- RGB LED
 - RED - PB4 - TIM3CH1
 - BLUE - PA9 - TIM1CH2
 - GREEN - PC7 - TIM3CH2
- 5-směrný joystick
 - PA4 = UP
 - PB0 = DOWN
 - PC1 = LEFT
 - PC0 = RIGHT
 - PB5 = CENTER
- Speaker – připojen na PB10 (= TIM2CH3)
- 2x potenciometr – piny PA0 a PA1
- LCD s rozlišení 128x32 (řadič ST7565R)
 - Připojen přes SPI, pouze signály SCK a MOSI (= nelze číst)
- Akcelerometr - MMA7660 – připojen přes I²C
- Teploměr – LM75B – připojen přes I²C



Úkoly na MBED shield

- Blikat složkami RGB LED
- Číst stav "tlačítek" joysticku
 - Vypisovat na UART – terminál v PC
 - Měnit "barvu blikání" LED
- Připravit si makra pro snazší využití
 - Např. soubor mbed_shield.h
 - Inspirace v nucleo_core.h
 - #define BOARD_LED GPIOA,5
 - Použito např. ve funkcích GPIOWrite a Nucleo_SetPinGPIO

Plán cvičení – aktivity v laboratoři

1. Úvod, rozdělení kitů, opakování samostatně
2. Pokročilé periférie ARM Cortex M
 - Timery, PWM
 - Využití LCD (SPI komunikace)
3. FPU - nastavení kompilátoru, porovnání rychlosti
4. Přerušení advanced - priority, blokování a řešení problémů
5. Problém atomických operací, řešení pomocí bit-banding
6. DMA - přenos bloku paměti, USART
7. DMA 2 - využití A/D převodníku a bufferu dat
8. Problematika Low-Power režimů
9. Privilegovaný a neprivilegovaný režim
10. SVC
11. Pokročilé techniky programování – zásobníky, semafore, instrukce LDREX/STREX, paměťové bariéry ...
12. RTOS - praktické řešení
13. Rezerva

Shrnutí domácí přípravy

- Máme připravené konstanty pro využití MBED Shield
- Umíme ovládat RGB LED
- Umíme číst stav joysticku
- Umíme komunikovat přes UART