

# KAE/MPP – cvičení

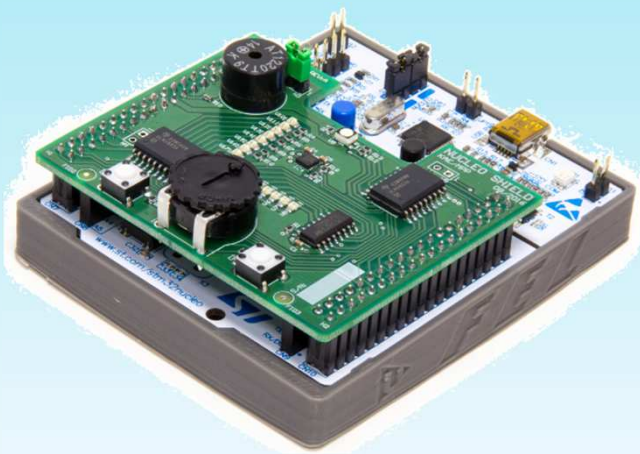
## Mikroprocesory a mikropočítače

**Verze 2016/17**

Ing. Petr Weissar, Ph.D.

Ing. Petr Krist, Ph.D.

Ing. Kamil Kosturik, Ph.D.



# Plán cvičení – aktivity v laboratoři

## 1. Úvod

- **seznámení s laboratoří, plán semestru**
- **distribuce vývojových kitů, ověření funkčnosti HW**
- **prostředí Atollic TrueStudio**

2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Plán domácí přípravy

- I. Instalace prostředí, ... První program – blikání GPIO
- II. Zobecnění GPIO operací
- III. SysTick
- IV. Timer
- V. UART
- VI. SPI
- VII. PWM
- VIII. A/D
- IX. I2C (využití "knihovny")
- X. Časování – HSI/HSE/PLL
- XI. Rezerva
- XII. Rezerva

# Organizace výuky, práce v laboratoři

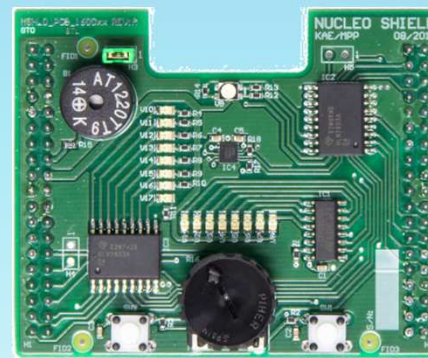
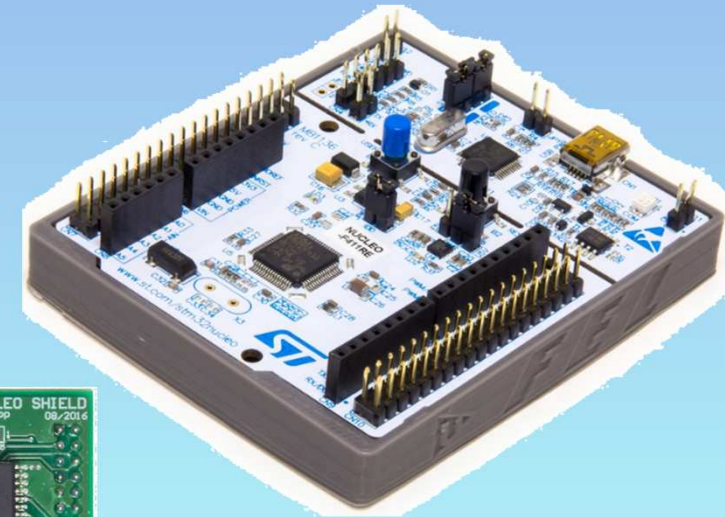
- V laboratoři pracujeme samostatně/ve dvojicích
  - Společný sdílený síťový adresář – disk T:
  - Domácí adresář studentů H:
  - Přilogování pomocí Orion-přístupu
  - Další síťové disky (R/O) - Z: (SW, podklady), X: (bat)
- HW společný – zapůjčen studentům
  - Předpokládá se část práce (příprava) doma
  - 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 minimálně 7 z 10 nutných úkolů na cvičení
  - Na každém cvičení několik úkolů, **vybrané jsou povinné**
  - Jejich splnění po vyzkoušení kroků z domácí přípravy by nemělo být komplikací
- 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 "Zapůjčce"
  - Jedna kopie zůstává, druhou má student
- Kit obsahuje v krabičce
  - Nucleo STM32F411RE
  - KAE/MPP 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](http://www.st.com), [mbed.org](http://mbed.org), ...
  - Vybrané na CourseWare nebo zde na Z:\podklady\MPP\...
- Podklady procesor:
  - **Reference manual** - RM0383 - STM32F411xE - rev 1 - DM00119316.pdf
    - Reference manual STM32F411xC/E advanced ARM®-based 32-bit MCUs
  - **Datasheet** - DS10314 - STM32F411xE - rev 6 - DM00115249.pdf
    - 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 - STM32F4 F3 L4 - rev 5 - DM00046982.pdf
    - STM32F3, STM32F4 and STM32L4 Series Cortex®-M4 programming manual
- Podklady Nucleo modul (<https://developer.mbed.org/platforms/ST-Nucleo-F411RE/>)
  - **User manual** – Nucleo UserManual rev10 - DM00105823.pdf
  - **Schematic** - Nucleo schematic - rev C - MB1136.pdf
- Podklady pro KAE/MPP 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>

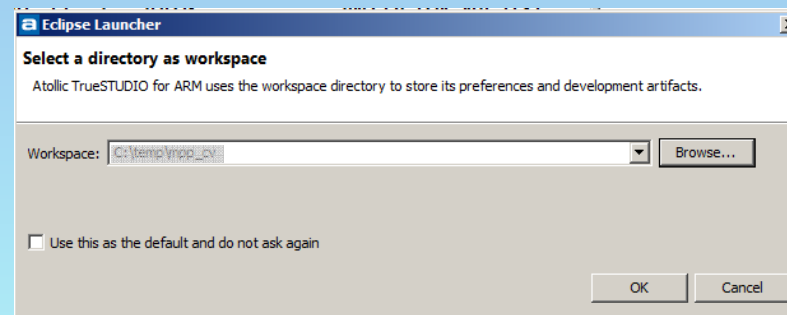
# Demo aplikace – ověření funkce HW

- Disk H: je "privátní" pro studenta, ověření Orion login
  - Teoreticky lze pracovat i ve skupině, ale budou se špatně realizovat samostatné cvičení
- Předpokládá se, že si studenti přinesou zpracované DCV
  - Buď na disku (USB-flash apod.) na naše PC
  - Nebo komplet vlastní notebook s funkčním IDE a svým Workspace
- První krok - vytvořit na H: pracovní adresář = "pracovní prostor" (Workspace)
- Nakopírovat do Workspace komplet adresář cv1
  - Nachází se na Z:\podklady\MPP\demo2017\cv1



# Spuštění prostředí Atollic TrueStudio

- Ikona na ploše: 
- Vybrat Workspace
  - = Vytvořený adresář na H:



- Import projektu
  - Nezbytné v případě prvního spuštění "prázdného workspace" nebo při nakopírování projektu do Workspace
    - Pravým tlačítkem v "Project Explorer" nebo v menu File/Import...
  - Pak si samozřejmě pamatuje rozpracované projekty
- Použité IDE založené na Eclipse
  - Projekty souborově orientované

# Kompilace a spuštění programu

- Ikony nebo menu

- Kompilace (build)



- Start Debug



- Pokud došlo ke změně v kódu, může se při spuštění Debug automaticky provést Build

- V Debug režimu počkat na nahrání programu do HW

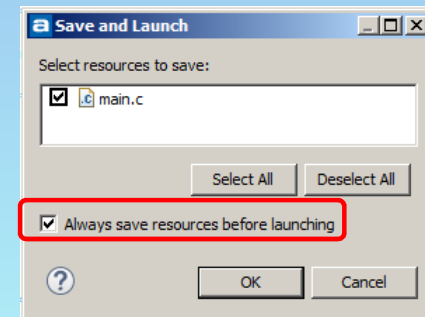
- Spustit příkazem Resume

- Ikona



- Menu Run/Resume

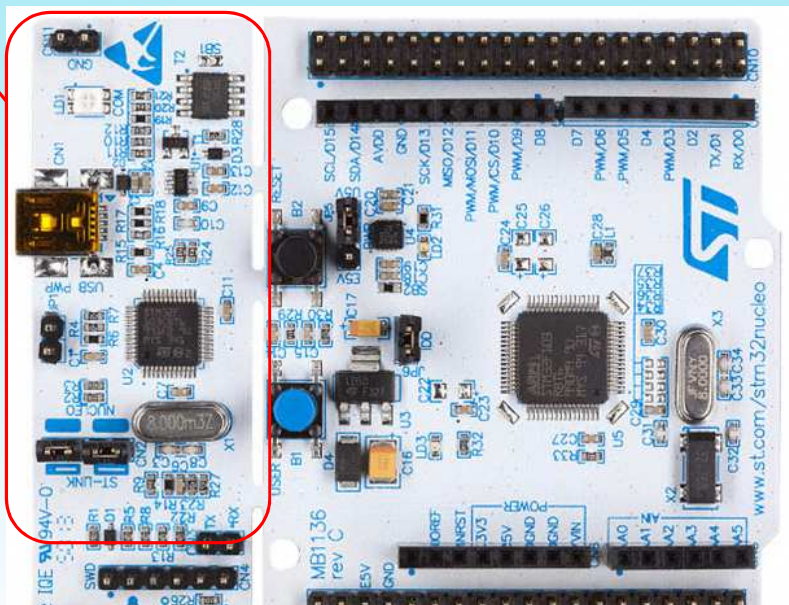
- Klávesa F8



- Bliká zelená LED na Nucleo (bílá) desce ...

# Představení vývojového kitu

- STM32 Nucleo – osazené STM32F411RE
  - Vybrané signály na "arduino" konektorech
  - Všechny signály na "morpho" konektorech
  - Připojení USB - obsahuje ST-link
    - Implementuje 3 "device" – Debug-SWD, USB/UART, MassStorage (USB)
    - Ovladače přímo od ST
      - ST-Link, ST-Link/V2, ST-Link/V2-1 USB driver signed for XP, Windows7, Windows8
      - ke stažení např. <http://www.st.com/en/embedded-software/stsw-link009.html>



- **KAE/MPP application shield** – pro Morpho konektory
  - Použijeme až od 2. cvičení
  - Zatím není nasazen



# Plán cvičení – aktivity v laboratoři

1. Úvod
2. **Procvičení práce s GPIO**
  - **Určení kam jsou GPIO připojeny (LED)**
  - **Ovládání více GPIO (LED) najednou**
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

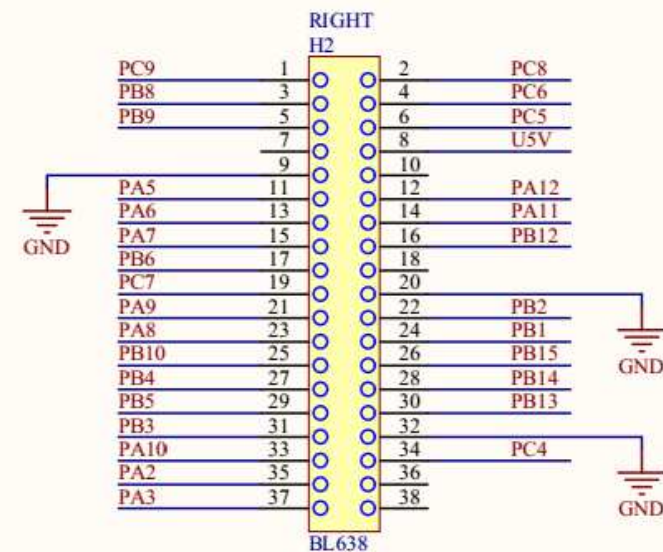
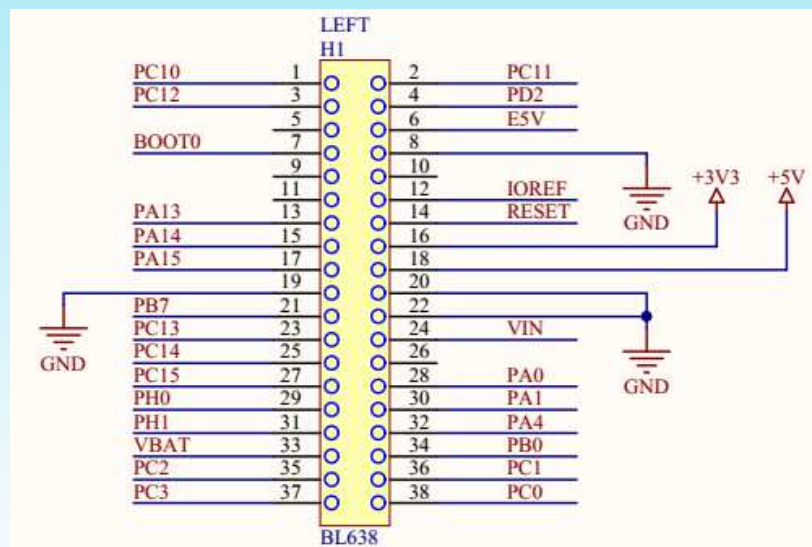
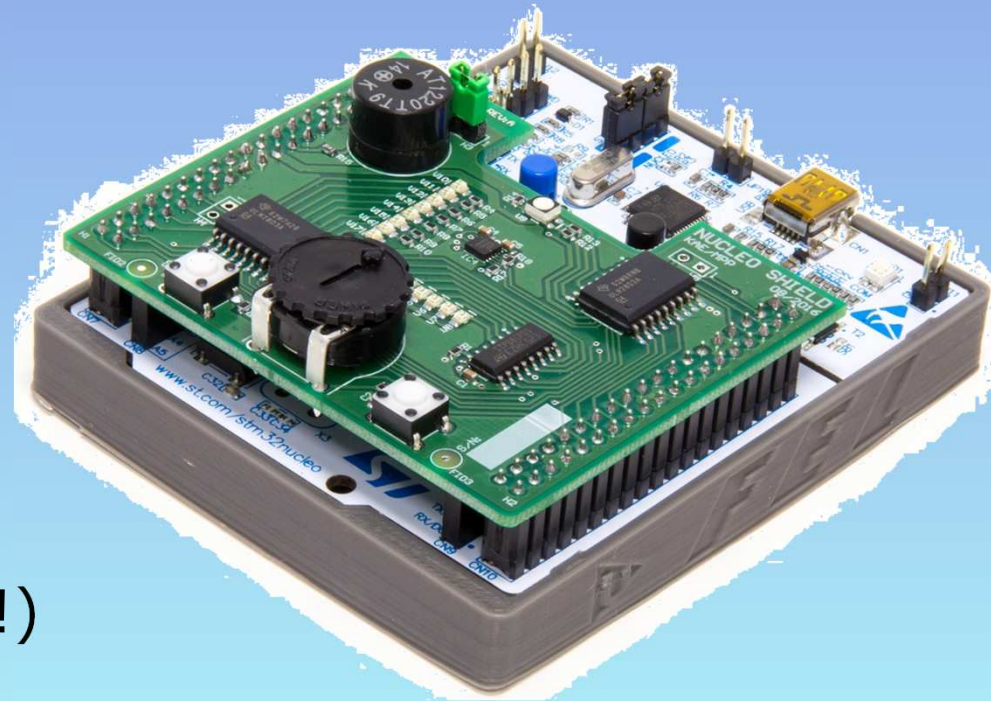
# Ověření domácí přípravy - 1

- Naučili jsme se
  - Založit Workspace a nový projekt pro STM32F411
  - Nastavit činnost GPIO
    - Povolit hodiny v RCC periférii
    - Nastavit režim příslušného výstupu
  - V nekonečném cyklu "blikat" LED-kou
    - Blikání pomocí bitového XOR v ODR registru
    - Čekání realizováno cyklem for
- Zopakovali jsme konstrukce jazyka C
  - Bitové operace – AND, OR
  - Maskování pomocí bitových operací
  - Posuvy a využití pro nastavení bitů v registrech pomocí konstant
  - Přístup k prvkům struktury přes ukazatel (její adresu)



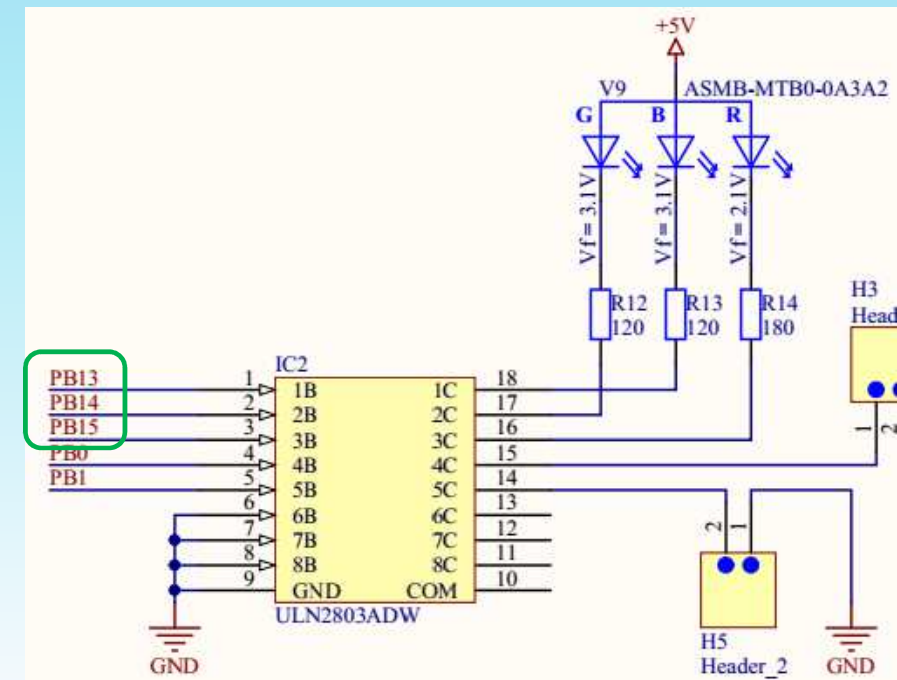
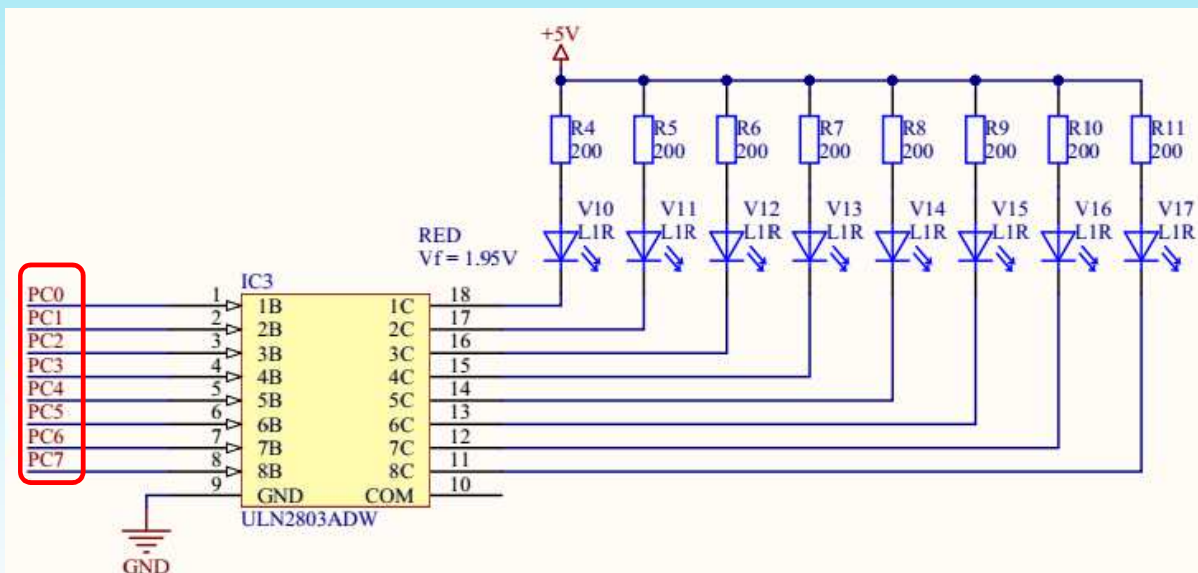
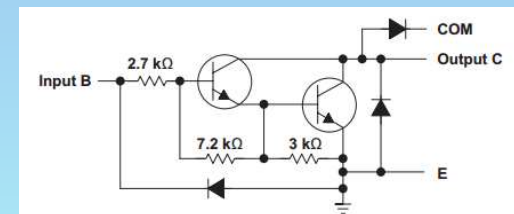
# KAE/MPP shield

- Určen pro Morpho konektory = využívá všech signálů procesoru
- Budeme už stále používat = nainstalovat (v odpojeném stavu !)
- Zůstává přístupný RESET a User button (modrý)



# 8x LED – červené a RGB LED

- Připojeny přes budič přímo na 8 signálů I/O brány
  - 8x červené na bránu **PC0-7 (GPIOC)**
  - LED RGB - složky na **PB13, PB14, PB15**
- Budič ULN2803A s Darlingtonovým zapojením tranzistorů
  - Nezatěžuje výstupy procesoru proudem LED
  - "Svítí" se log. 1



# Procvičení práce s GPIO

- V domácí přípravě jsme "blikali" LED na PA5
- **Upravte kód pro využití jedné z červených LED**
- Upravte kód pro blikání více červených LED
- Upravte kód pro blikání více LED různou rychlostí
- **Upravte kód pro blikání jednou složkou RGB LED**
- Upravte kód pro blikání jednotlivými složkami RGB LED
  - Lze vytvořit 7 barev mícháním (8. barva je "zhasnuto")



# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. **Procvičení pokročilé práce s GPIO**
  - **Varianty efektu typu běžící bod na 8xLED**
  - **Interakce s uživatelem pomocí tlačítek**
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

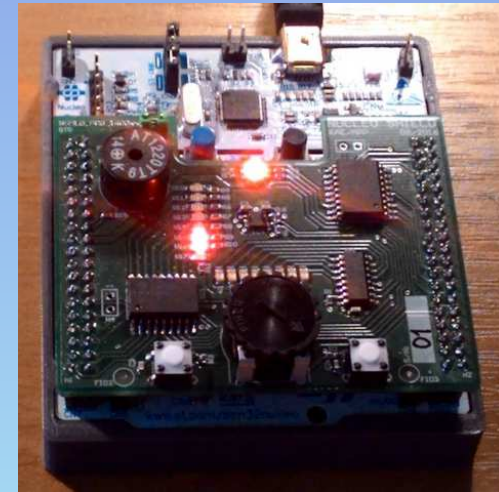
# Ověření domácí přípravy - II

- Umíme nastavit všechny základní režimy GPIO
  - Máme připravenou univerzální funkci pro korektní nastavení jednoho IO pinu
- Máme připravenou funkci pro nastavení výstupu a čtení vstupu
- Máme připravenou funkci pro nastavení GPIO pro 8xLED
  - Připojené na PC0-7
  - Nastavují se najednou
- Máme připravené makro pro nastavení hodnoty PC0-7
- Funkce a makra máme v samostatných souborech (názvy podle ukázky)
  - Pro procesor obecně v **stm\_core.c** a **stm\_core.h**
  - Pro "náš" KAE/MPP shield v **mpp\_shield.c** a **mpp\_shield.h**
- Umíme připojit do svého main.c pomocí #include
- Máme nadefinovaná makra pro RGB LED a tlačítka

# Kopírování projektů a souborů mezi projekty

- Projekt lze klasicky kopírovat
  - Pravým klikem na projektu – Copy, pravým ve volném místě "Project Explorer" – Paste
  - Automaticky se pokusí navrhnout nové jméno projektu
  - POZOR, nejde kopírovat ze "zavřeného projektu"
- Pokud máme otevřeno více projektů (ve Workspace), je možné kopírovat soubory (typicky z SRC adresáře)
  - Přetáhnutím myši se stisknutým CTRL (u šipky je "plus")
  - Klasicky CTRL+C, CTRL+V
  - Pravým klikem na souboru – Copy, následně na cílovém adresáři – Paste

# Procvičení pokročilé práce s GPIO



- V dcv jsme si na 8xLED ukázali vzor "počítadlo"
- Vytvořte program, který na 8xLED ukáže efekt "běžící bod"
  - Zřejmě bude potřeba použít operace posuvu
- Upravte program, aby efekt běžel v opačném směru
- **Upravte program, aby efekt měnil směr (ping-pong)**
- Upravte program, aby se efekt dal zastavit podržením tlačítka
- **Upravte program, aby se efekt dal zastavit jedním stiskem tlačítka a dalším stiskem opět rozběhl**
  - Bude potřeba použít minulý a současný stav tlačítka detekovat změnu (obdobu klopného obvodu v číslicovce)
- Upravte program, aby při "odražení" bliknula jedna ze barev RGB LED

# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. **Procvičení časování pomocí SysTick a přerušení**
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Ověření domácí přípravy - III

- Umíme používat SysTick
  - Nejčastěji v režimu s přerušáním
  - Každým "tikem" se inkrementuje počítadlo "tiků"
  - V aplikaci je možné ho použít jako časovou základnu
- Umíme se podívat v debuggeru na hodnotu SystemCoreClock
- Umíme hodnotu SystemCoreClock použít pro nastavení SysTick
- Umíme jednoho časového základu využít pro měření různých intervalů

# Procvičení práce se SysTick

- Použijte SysTick jako časovou základnu s "tikem" 1ms
- Vytvořte program, který bude blikat jednotlivými složkami RGB LED v různém rytmu (např. 50, 75, 100 ms)
- **Upravte program tak, aby bylo možné blikání jednotlivých RGB složek blokovat tlačítkem**
  - Na Shieldu jsou 2 tlačítka, jako třetí použijte modré tlačítko na Nucleo desce (připojené na PC13)
- Vytvořte program, který bude blikat jednotlivými LED z 8xLED, každá jinou rychlostí
  - Časy nejsou navzájem jednoduché násobky

# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
- 5. Procvičení – práce časovačem (TIM3) - generování tónu i časování**
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty



# Ověření domácí přípravy - IV

- Umíme nastavit čítač/časovač v základním režimu
  - Vyzkoušeno na TIM3
- Umíme otestovat příznak přetečení/podtečení
- Umíme použít přerušení od přetečení/podtečení
  - Nastaven základní režim NVICu
  - Nezapomenout "shodit" příznak události
- Umíme generovat tón na připojeném "pípáku" využitím přerušení
- Připomněli jsme si význam proměnné typu static
  - Použito pro počítání počtu vyvolání přerušení

# Procvičení práce s časovačem

- **Vygenerujte na výstupu pro "pípák" kmitočet přesně 1kHz a ověřte pomocí osciloskopu**
- Pomocí jednoho tlačítka opakovaným stiskem zapínejte/vypínejte generování tónu "pípákem"
- Pomocí druhého tlačítka měňte výšku tónu (výběr alespoň ze 2)
- Použijte jiný časovač – např. TIM4 a v "jeho" přerušení blikajte barvou LED nezávisle na TIM3
- **Nastavte TIM3 tam, aby v registru CNT (samotný obsah čítače) měl pouze 8-bitovou hodnotu**
  - **Tuto hodnotu "kopírujte" do 8xLED** – nejlépe se pro to hodí samotná nekonečná smyčka – nevadí, když se vícekrát zobrazí "stejná" hodnota
  - Zřejmě bude nutno zvolit co nejvyšší prescaler (PSC registr), aby se obsah CNT měnil co nejpomaleji
- Doplňte zobrazování obsahu čítače o možnost změnit směr počítání tlačítkem

# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
- 6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO**
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Ověření domácí přípravy - V

- Umíme nastavit USART do režimu asynchronního přenosu
  - Používáme USART2, protože "jeho" GPIO jsou připojeny na ST-Link, který realizuje virtuální COM port přes USB
  - Umíme vypočítat dělicí poměry v BRR registru pro danou komunikační rychlost a takt hodin
- Umíme otestovat příznaky při příjmu a vysílání
- Máme připravené funkce zabalující základní operace
- Umíme v terminálu na PC přijmout i odeslat znaky
- Umíme využít funkce z knihovny STDIO
  - Máme vyzkoušené funkce `_read` a `_write` v `stdcalls.c`

# Procvičení práce se sériovou komunikací

- Na 8xLED zobrazte kód přijímaných znaků z PC
- Při příjmu znaku vypište zpět na terminál ASCII kód znaku
- **Při stisku tlačítka na desce vypište na terminálu zprávu, které to bylo tlačítko**
- **Umožněte rozsvícení složek RGB LED podle pokynů z terminálu – např. ve formě znaků**
  - R rozsvítí, r zhasne červenou složku LED
  - G rozsvítí, g zhasne zelenou složku LED
  - B rozsvítí, b zhasne modrou složku LED

# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
- 7. Procvičení probraných částí, vyrovnání neprobraného**
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Procvičení probraných částí

- Co vše je zatím probráno:
  - GPIO
    - Režim vstupní – PullUp a PullDown rezistory
    - Režim výstup – PushPull režim (případně OpenDrain)
    - Alternativní funkce
    - Registry – datové ODR, IDR, BSRR; nastavovací MODER, OTYPER, ...
  - SysTick
    - Časování s milisekundovou základnou
  - Přerušení a NVIC
    - Povolení IRQ, nalezení jména Handleru, realizace funkce
    - "Shození" požadavku přerušení v periférii
  - Časovače (např. TIM3)
    - Registry CRx, PSC, ARR, CNT, SR
  - Sériový port (USART2)
    - Nastavení rychlosti BRR, registry CRx, status-bity TXE, TC, RXNE
    - Využití std. funkcí knihovny STDIO – úprava souboru syscalls.c

# Úkoly

- Upravte program pro efekt "běžící světlo" na 8xLED tak, aby podle odeslaných povelů (znaků) z PC terminálu šlo
  - Změnit směr nebo zastavit efekt v poslední poloze a spustit
  - Změnit rychlost v několika přednastavených krocích
- Nastavte frekvenci generování tónu zadáním číselné hodnoty na terminálu
  - Ověřte hlavně na osciloskopu
  - Funkce scanf z STDIO nefunguje správně, proto ji nepoužívejte
- Realizujte program typu "piáno"
  - Zvolená písmena z klávesnice odpovídají tónům
  - Minimálně 8 tónů stupnice
  - Frekvence např. obrázek na <http://www.agadir.cz/teorie.php?vyber=1>
- Vypište obsah bloku paměti
  - Nastavte adresu začátku
  - Vypisujte po řádcích, na začátku adresa, následně 8 nebo 16 hodnot, vše hexadecimálně
  - Vypište buď počet nebo znakem z terminálu volte "další" řádek nebo konec výpisu



# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
- 8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED**
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Ověření domácí přípravy - VII

- Prozkoumali jsme datasheet k budiči LED se sériovým výstupem
  - Dokážeme vyčíst potřebu generovaných signálů
  - Dokážeme najít časové informace o průběhu signálů včetně max. frekvence hodin
- Umíme generovat průběh podobný SPI programově pomocí GPIO operací
- Umíme odpovídající průběhy generovat pomocí bloku SPI
  - Umíme nastavit CRx registry – např. fáze a polarita
  - Umíme využívat příznaky ve stavovém registru SR
- Máme připravené funkce pro inicializaci a zápis dat s možností změnit přístup přes GPIO nebo přes SPI

# Procvičení práce s SPI (budič 8x LED)

- Připojte osciloskop
  - Na signály SCK a MOSI z procesoru a sledujte průběhy
    - Odpovídají signálům CLK a SDI budiče LED
  - Sledujte také signál LE budiče
  - Pro jednoduchost není nutné odpojovat MPP-Shield, zapůjčíme náhradní Nucleo desky a měřit se bude na nich
- **Porovnejte průběhy při generování výstupů kódem s GPIO a využitím SPI**
- Ověřte chování při změně testu SPI "odvysílání" následované LE pulsem
- **Realizujte efekt běžícího bodu s 16 prvky**
  - **Použijte 8xLED na GPIOC a 8xLED na SPI budiči**
  - Hodnotu efektu mějte v jedné proměnné uint16\_t
  - Demo efektu např. na <https://youtu.be/SWbfGTYxxMw>

# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
- 9. Procvičení - pomocí PWM generování barev na RGB LED**
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Ověření domácí přípravy - VIII

- Umíme použít CAPCOM jednotku čítače/časovače pro generování
  - Signálu se střídou 50% a proměnnou frekvencí/periodou
    - Ověřeno na výstupu pro "pípák"
  - Signálu s fixní periodou a proměnnou střídou
    - Ověřeno na složkách RGB LED
      - Postupné rozsvěcení
      - Nastavení libovolné barvy (např. v debug perspektivě při zastaveném programu)
- Viděli jsme, každý čítač má až kanály s CAPCOM funkcionalitou
- Viděli jsme, že TIM1 má některé speciální vlastnosti, které musíme zohlednit v programu

# Procvičení práce s PWM

- Připojte osciloskop
  - Na vývod PB0 (= "pípák") a sledujte průběhy
    - Generujte na výstupu PWM z TIM3 kanálu 3
- **Sledujte průběhy – v debug režimu při zastaveném programu měňte obsah ARR a CCR3 registrů**
- **Realizujte efekt "dýchání barvy" na RGB LED**
  - **Umožněte změnu barvy**
  - Buď pomocí tlačítek nebo přes terminál z PC
  - "dýchání" znamená pozvolné rozsvěcení a zhasínání
- Nastavte libovolnou RGB barvu pomocí terminálu

# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
- 10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO**
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Ověření domácí přípravy - IX

- Umíme nastavit A/D převodník tak, aby měřil z externího vstupu
  - Spouštění převodu programově
  - Testování dokončení převodu také programově
  - Využíváme "regular" režim a jednorázové měření
- K ověření činnosti využíváme potenciometr připojený na vstup AD0 (odpovídá pinu PA0)



# Procvičení práce s AD převodníkem

- **Změřte dobu převodu pro různá nastavení SMP0 v registru SMPR**
  - Nastavte PB0 jako výstupní
  - Připojte osciloskop na vývod PB0 (= "pípák") a sledujte průběhy
  - Před spuštěním převodu nastavte log. 1, po dokončení (EOC příznak) nastavte log. 0
    - Šířka pulsu odpovídá době převodu
- **Realizujte efekt "bargraf" na 8xLED**
  - Podle hodnoty z AD převodníku rozsviďte sloupec LED = čím větší hodnota, tím více LED svítí
- **Měňte hodnotu PWM (= jas barvy RGB LED) podle AD hodnoty potenciometru**
  - Vhodně zvolte počet hodnot PWM tak, aby se hodnota pro střidu snadno použila z měření AD
- **Měřte analogovou hodnotu z teplotního senzoru**
  - Povolit TSVREFE v ADC\_RCC registru
  - Měřit z kanálu 16 (bity SQ1 v SQR3)
  - Viz. RM kap. 11.9, "Reading the temperature"
    - Teplota podle vzorce  $\text{Temp [}^\circ\text{C]} = \{(V_{\text{MERENE}} - V_{25}) / \text{Avg\_Slope}\} + 25$
    - Hodnoty  $V_{25}$  a Avg\_Slope viz. DS kap. 6.3.21 – str. 119/149

# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
- 11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"**
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
13. Rezerva, zápočty

# Ověření domácí přípravy - X

- Naučili jsme se pracovat s komunikační sběrnici I2C
  - Využili jsme předpřipravenou knihovnu (modul) pro naši aplikaci
  - Prozkoumali jsme základní registry periférie I2C v mikrokontroléru i princip samotné sběrnice
- Využili jsme připojený vnější obvod – akcelerometr MMA7660
  - Je 3-osý s citlivostí  $\pm 1,5g$ , 6-bitová hodnota pro každou osu
  - Umíme přepočítat změřenou hodnotu z vnitřních registrů akcelerometru na fyzikální veličinu – zrychlení
  - Pomocí hodnoty zrychlení ve všech osách umíme určit polohu zařízení vůči zemi

# Procvičení práce s akcelerometrem na I<sup>2</sup>C

- **Vyčítejte data z akcelerometru jako blok registrů 0x00 – 0x03 a vhodně zobrazujte**
  - Funkce **I2C1\_ReadBytes**
  - Reprezentují hodnotové XOUT, YOUT, ZOUT
  - Registr TILT je bitově orientovaný – vypište jednotlivé významy
- **Využijte X- a Y-osu pro simulaci 2D vodováhy**
  - Osa X vytváří "efekt svítícího bodu" na červených LED
  - Osa Y vytváří stejný efekt na zelených LED
  - Celkový efekt – v klidu svítí "prostřední" LED, při naklápění dle X nebo Y osy se svítící LED posouvá podobně jako bublinka vodováhy
- Využijte aplikaci na PC zobrazující data z akcelerometru
  - Pro absolventy KAE/PPES si přineste známý program
  - Pro ostatní je EXE na "podkladech"
  - Očekávaný formát:
    - Řádek ukončený \r nebo \n (nebo obojí)
    - Data oddělená ;
    - Hodnoty celočíselné v tisících G se znaménkem, tj. 1g = 1000, -½g = -500

# Plán cvičení – aktivity v laboratoři

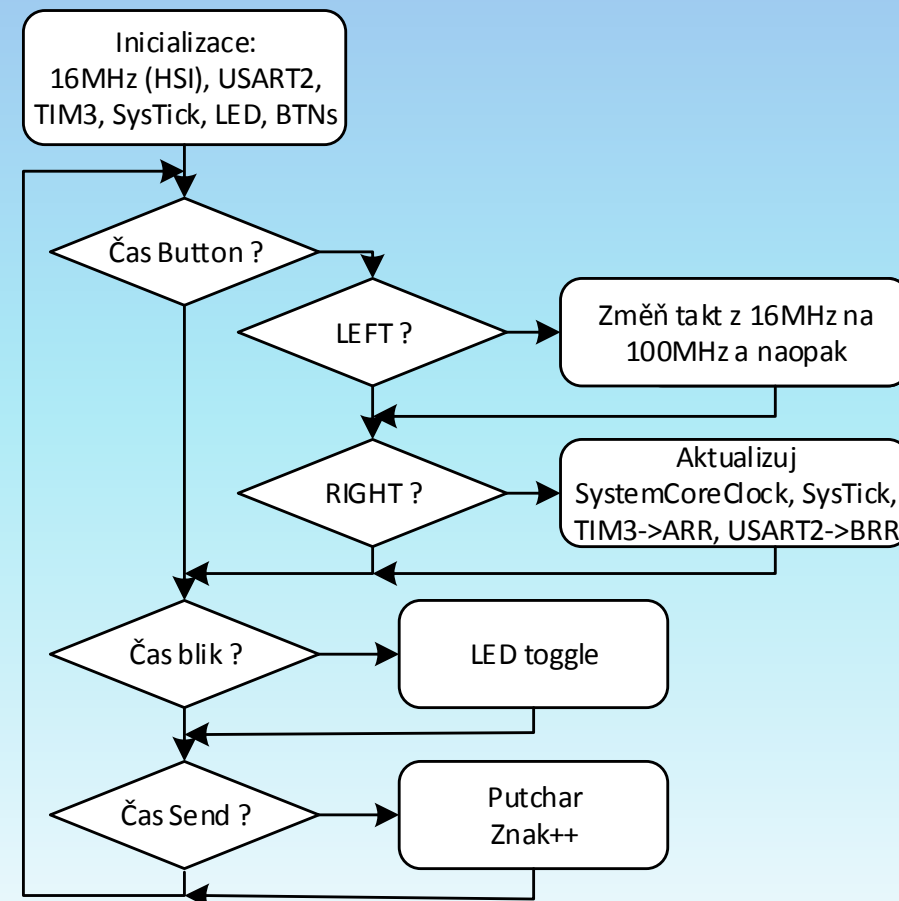
1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
- 12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)**
13. Rezerva, zápočty

# Ověření domácí přípravy - XI

- Máme připravené funkce pro nastavení základních hodin
  - Varianta 100MHz z externího HSE
  - Varianta 16MHz z HSI (defaultní stav po RESETu)
- Máme připravené funkce pro zjištění taktu sběrnice AHB a APBx a umíme je použít v nastavení periférií – např. TIMx
- Máme upravenou funkci inicializace USART tak, aby "si spočítala" hodnotu děliče BRR podle aktuálních hodin APBx a požadované komunikační rychlosti

# Procvičení práce s blokem RCC

- Vytvořte aplikaci, která:
  - Využívá k 1ms časování SysTick
  - Bliká LED (např. jedna z 8xLED červených)
  - Umí zjistit stisk LEFT a RIGHT tlačítka
  - Pravidelně odesílá sériově znaky do PC
  - "Píská" v taktu 1kHz na výstupu PB0
  - V základu běží na 16MHz z HSI (RESET-state)
- Po stisku LEFT změní časování
  - Mezi 100MHz z HSE a 16MHz z HSI
- Po stisku RIGHT aktualizuje takty periférií
  - SysTick, TIM3, USART2
- **Sledujte vizuálně a na osciloskopu**
  - **Změnu rytmu blikání LED**
  - **Změnu tónu/frekvence na PB0**
- **Porovnejte přesnost 1KHz z HSI a HSE**



# Plán cvičení – aktivity v laboratoři

1. Úvod
2. Procvičení práce s GPIO
3. Procvičení pokročilé práce s GPIO
4. Procvičení časování pomocí SysTick a přerušení
5. Procvičení – práce časovačem (TIM3) - generování tónu i časování
6. Procvičení - sériová komunikace (UART) s PC s spojení s GPIO
7. Procvičení probraných částí, vyrovnání neprobraného
8. Procvičení - komunikace přes SPI - spojení s GPIO - řízení LED
9. Procvičení - pomocí PWM generování barev na RGB LED
10. Procvičení - použití A/D převodníku, spojení s PWM a GPIO
11. Procvičení - komunikace s akcelerometrem přes I2C - aplikace "vodováha"
12. Ověření výběru zdroje hodin a nastavení taktu procesoru (např. pomocí osciloskopu)
- 13. Rezerva, zápočty**