STM32CubeIDE, Git, základní funkce

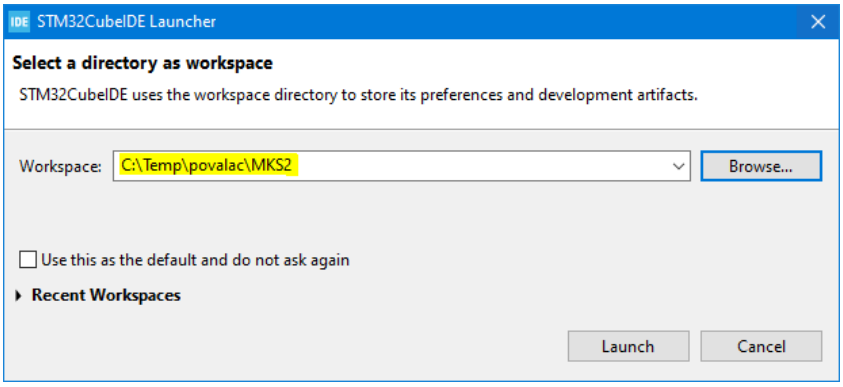
1 Zadání

- Seznamte se s vývojovou deskou NUCLEO-F030R8, modulem STMrel a prostředím STM32CubeIDE.
- Seznamte se s programem TortoiseGit. Založte si repozitář na Githubu, vytvořte pracovní kopii adresáře (Git Clone).
- Vytvořte jednoduchou aplikaci, která bude blikat s diodou LD2. Otestujte na vývojové desce, ověřte ladění krokováním. Uložte vytvořenou aplikaci do repozitáře (Git Commit).
- Vytvořte aplikaci, která bude realizovat blikání diodou LD2 dle předdefinovaného řetězce v morseovce. Commitujte do repozitáře. Následně kód upravte tak, aby byla sekvence morseovky zapsaná v jediné 32bitové konstantě, commitujte do repozitáře.
- Lokální repozitář uložte na server (Git Push).

2 Návod

2.1 Verzování a TortoiseGit

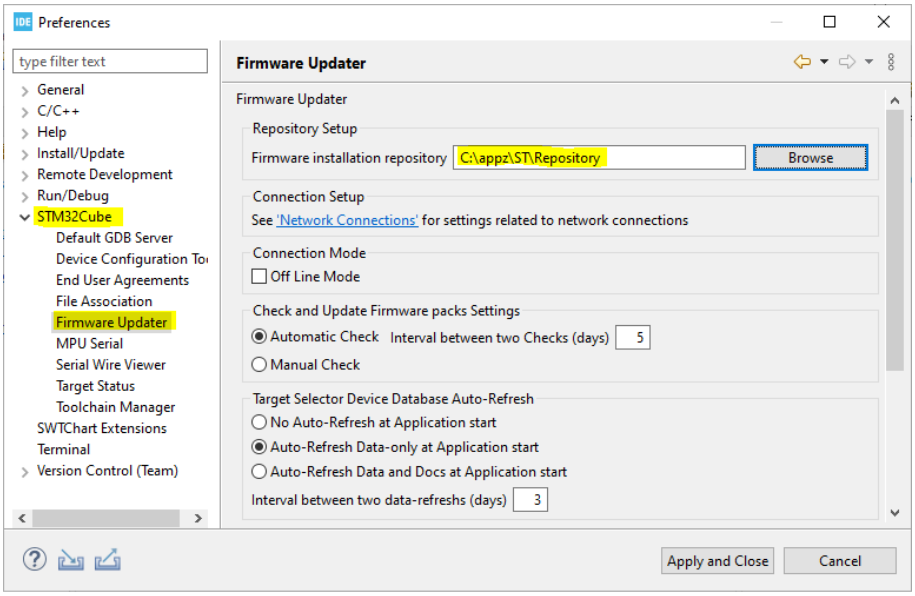
- Založte si účet na Githubu, vytvořte si nový repozitář nazvaný např. MKS.
- Ve složce **C:\Temp** si založte svůj pracovní adresář a v něm vytvořte pomocí operace Git Clone pracovní kopii svého osobního repozitáře. Pro přihlášení na Github využijte možnosti Sign in with your browser.
- Do této pracovní kopie přepněte workspace STM32CubeIDE (při startu nebo pomocí File / Switch Workspace).



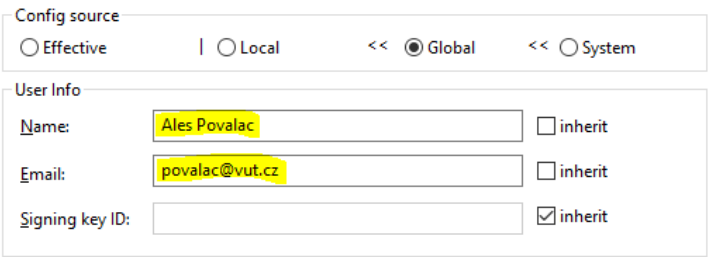
- Dále pracujte pouze na pracovní kopii.

2.2 Základní seznámení

- Potřebná schémata vývojové desky najdete v eLearningu. Modul STMrel v tomto cvičení nevyužijeme.
- Prostředí STM32CubeIDE ukládá množství dat lokálně, zejm. repozitáře s knihovnami. Aby nebylo nutné znovu vše stahovat, je vhodné nastavit cestu k centrálnímu repozitáři knihoven (platí pro učebny, ne soukromá PC):



- Nový projekt s HAL knihovnami se zakládá přes File / New / STM32 Project / Board Selector / NUCLEO-F030R8.
- Pro prvních několik cvičení ale budeme využívat přímého přístupu k registrům a příkladů z STM32SnippetsF0. V tomto případě změníme Targeted Project Type na Empty.
- V projektech budeme využívat CMSIS (Cortex Microcontroller Software Interface Standard):
  - Do projektové složky okopírujte z STM32SnippetsF0 (ke stažení v eLearningu) adresář CMSIS (stačí Device a Include).
  - Přidejte mezi prohledávané cesty: Project / Properties / C/C++ Build / Settings / Tool Settings / MCU GCC Compiler / Include Paths, přidat z projektu CMSIS/Device/ST/STM32F0xx/Include a CMSIS/Include.
  - Definujte správně symbol procesoru: stejný dialog / Preprocessor, přidat STM32F030x8.
  - Nakonec okopírujte soubor s inicializací systému: CMSIS/Device/ST/STM32F0xx/Source/Templates/system_stm32f0xx.c do složky Src.
- Překlad (Build all) lze spustit pomocí zkratky Ctrl+B, ladění (tj. spuštění, Debug) projektu pomocí F11.
- Při prvním spuštění se zobrazí nastavení debuggeru, potvrďte výchozí konfiguraci. Pokud je firmware debuggeru v připojené vývojové desce neaktuální, může být vyžádán upgrade.
- Během krokování ladění lze s výhodou používat klávesové zkratky, viz menu Run.
- Uložení pracovní kopie do repozitáře proveďte pomocí Git Commit. Git poprvé vyzve k zadání jména a emailu.



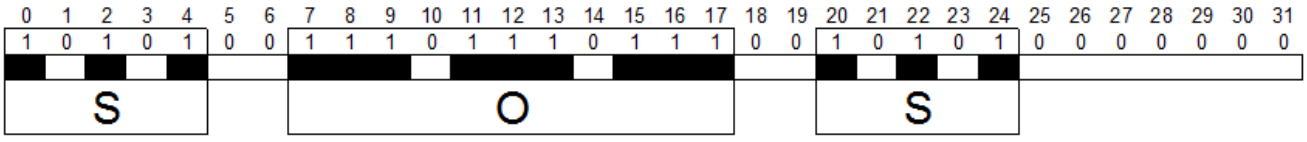
- Commitujte pouze zdrojové kódy (c, h, Makefile apod.), nikoliv výsledky překladu (o, d, elf, hex, soubory workspace apod.). S výhodou lze použít ignore listů, základní .gitignore jsou vytvořeny prostředím (ignorování složky .metadata a v jednotlivých projektech složek Debug).
- Uložte lokální repozitář na server (Git Push). Prohlédněte si výsledek přes web Githubu.

2.3 Rozblikání LED

- Ohledně přístupu ke GPIO se můžete inspirovat v Projects/GPIO/01_LockingMechanism.
- Je třeba includovat soubor stm32f0xx.h.
- Vývoj začněte funkcí main(). Inicializujte periferie, do nekonečné smyčky vložte negaci příslušného pinu a krátké čekání. Veškeré níže uvedené registry si ověřte v referenčním manuálu procesoru.
- LED dioda LD2 je připojená na PA5, který musí mít zapnuté hodiny a být nastaven jako výstupní (mód 01):
  - `RCC->AHBENR |= RCC_AHBENR_GPIOAEN;`
  - `GPIOA->MODER |= GPIO_MODER_MODER5_0;`
- Změny výstupní úrovně pinu je možné provádět zápisem do registrů BSRR a BRR, příp. ODR:
  - `GPIOA->BSRR = (1<<5); // set`
  - `GPIOA->BRR = (1<<5); // reset`
  - `GPIOA->ODR ^= (1<<5); // toggle`
- Čekání lze řešit jednoduchou busy-wait smyčkou:
  - `for (volatile uint32_t i = 0; i < 100000; i++) {}`

2.4 Blikání v morseovce

- Vývojová deska bude pomocí LD2 blikat v Morseově abecedě kód „SOS“.
- Sekvenci můžete uložit jako jedničky (svítí) a nuly (nesvítí) např. do pole typu `uint8_t pole[32]` a provést jeho inicializaci přímo v kódu.



- Kromě samotného pole budete potřebovat iterační proměnnou, která bude do tohoto pole ukazovat. Bude-li mít pole na indexu dle této proměnné hodnotu 0, výstup pro LD2 nastavíme na log. 0, v případě nenulové hodnoty na log. 1.
- Proveďte commit pracovní kopie do Gitu.
- Optimalizace využití paměti: Sekvence bude uložená v typu `uint32_t`, využijte binární zápis prvků (např. `0b0100`), dosáhněte stejné funkcionality. Budete potřebovat bitové operace a bitové posuny.
- Proveďte commit pracovní kopie do Gitu.
- Nezapomeňte odeslat lokální repozitář na server (Git Push).