Představení laboratorního kitu

IMP – demo cvičení

Ing. Václav ŠIMEK

Vysoké učení technické v Brně, Fakulta informačních technologií Božetěchova 1/2, 612 66 Brno, Česká republika simekv@fit.vutbr.cz



30.09.2022

Obsah



- Výukový kit s 32-bit ARM MCU
- Schéma zapojení a popis periferií
- Vývojové prostředí KDS
- Praktická ukázka kódu

Výukový kit s 32-bit ARM MCU – finální verze





V této podobě je kit nainstalován v laboratoři L306

Schéma zapojení – ladicí rozhraní OpenSDA







Schéma zapojení – mikrokontrolér MKL05Z32





Schéma zapojení – sériová linka RS232





Schéma zapojení – mechanická tlačítka





Schéma zapojení – piezzo bzučák a RGB led



FIT





Vývojové prostředí KDS – základní ovládací prvky





- 1) funkce pro založení projektu a jeho základní konfiguraci
- 2) překlad projektu
- 3) nastavení ladicího rozhraní a nahrání výsledné binárky do MCU
- 4) přepínání mezi editací kódu a laděním (*C/C++ perspective, Degug perspective*)

Vývojové prostředí KDS – postup při založení projektu





- 1) v menu "File" zvolíme položku NEW a dále Kinetis Project
- 2) do kolonky "*Project name*" zadáme název projektu
- 3) v menu "Devices" vybereme požadovaný typ mikrokontroleru, tj. MKL05Z32

Vývojové prostředí KDS – konfigurace ladicího rozhraní 🛛 🔽 F

Create, mana	ge, and run configurations				Ť
Image: Second secon	Application Attach to Application Postmortem Debugger Remote Application ardware Debugging EMicro Interface Debugging Debug_PNE reni_tepu_LED_disp_test_Debug_PNE IGGER J-Link Debugging Group	Name: aaa_Debug_PNE Main ** Debugger Project: aaa C/C++ Application: Debug/aaa.elf Builton-quired) before launching Builton-figuration: Use Active O E auto build O L dspace settings	Variables	Search Project)	Erowse

- **1)** ve sloupci vlevo najdeme rozbalovací menu *"GDB PEMicro Interface Debugging",* v němž klikneme na položku s názvem našeho projektu
- zkontrolujeme, zda řádek "Project" obsahuje název našeho projektu a zda je na řádku "C/C++ Application" uvedena cesta k výsledné binárce
- **3)** přejdeme na záložku "Debugger" a dokončíme nastavení ladicího rozhraní pro náš projekt

Vývojové prostředí KDS – konfigurace ladicího rozhraní 🛛 🔽

Ebug Configurations	×
Create, manage, and run configurations Plugin has not been registered. Some functionality magnetic series of the series of th	iy not be available.
Image: Second	Name: aaa_Debug_PNE Main Statup PNE Main Statup Power Software Registration Please register your software to remove this message. Register now PEMicro Interface Settings Interface: OpenSDA Embedded Debug - USB Port Port: USB 1 - OpenSDA (F7337E0A) Select Device Vendor: Freescale Family: K6x Target: K60DN512ZM10 Specify IP Specify Network Card IP Additional Options Mass erase on connect Hardware Interface Power Control (Voltage> Power-Out Jack) ms Provide power to target Regulator Output Voltage Power off target upon software exit ZV Power Down Delay Target Communication Speed Debug Shift Freq (KHz) ⁰ 5000 Oms Delay after Reset and before communicating to target for 0 ms ms
Filter matched 18 of 19 items	Apply Revert
?	<u>Debug</u> Close

- zkontrolujeme, že řádek "Interface" obsahuje správný typ debuggeru pro náš projekt, tj. <u>OpenSDA Embedded Debug – USB Port</u>
- 2) dále by měla být na řádku "Port" vidět konkrétní instance připojeného debuggeru, který se nachází na laboratorním kitu
- 3) pokud jsou všechna nastavení v pořádku, můžeme proces nahrávání kódu do MCU a jeho ladění zahájit kliknutím na ikonku "Debug"

Vývojové prostředí KDS – spuštění nahraného kódu





- 1) zelená šipka spustí výsledný kód našeho projektu, který byl nahrán do MCU
- 2) tlačítko "pauza" pozastaví provádění kódu, pro pokračování provádění kódu naší aplikace klikneme na zelenou šipku
- 3) pokud již chceme ukončit ladění aplikace, je třeba kliknout na červený čtvereček a ověřit, že se jeho barva změnila na "šedivou", jinak nepůjde ladění opět spustit
- 4) po ukončení ladění klikneme na pohled "C/C++" a přepneme se zpět do editačního režimu

Ukázka kódu – ovládání piezzo bzučáku

```
// Just an ordinary delay loop
#include "MKL05Z4.h"
void delay(long long bound) {
    long long i;
    for(i=0;i<bound;i++);</pre>
}
int main(void) {
    SIM->COPC = SIM_COPC_COPT(0x00); // Just disable the usage of WatchDog feature
    SIM->SCGC5 = SIM_SCGC5_PORTB_MASK; // Turn on clocks for PORTA and PORTB
    PORTB->PCR[13] = ( 0|PORT PCR MUX(0x01) ); // Set corresponding PORTB port pins as outputs
    PTB->PDDR = GPIO PDDR PDD( 0x2000 ); // "1" configures given pin as an output
    for (;;) {
         PTB->PDOR = GPIO PDOR PDO(0x2000);
         delay(500);
         PTB->PDOR = GPIO PDOR PDO(0x0000);
         delay(500);
     }
     return 0;
}
```

Vývojové prostředí KDS – názvy přerušovacích rutin



/* External Interrupts*/

DMA0 IRQHandler .long DMA1 IRQHandler .long DMA2 IRQHandler .long .long DMA3 IRQHandler Reserved20 IRQHandler .long .long FTFA IRQHandler LVD LVW IRQHandler .long .long LLWU IRQHandler .long I2C0 IRQHandler .long Reserved25 IRQHandler SPI0 IRQHandler .long Reserved27 IRQHandler .long .long UARTO IRQHandler .long Reserved29 IRQHandler Reserved30 IRQHandler .long ADC0 IRQHandler .long .long CMP0 IRQHandler .long TPM0 IRQHandler TPM1 IRQHandler .long .long Reserved35 IRQHandler .long RTC IRQHandler .long RTC Seconds IRQHandler PIT IRQHandler .long Reserved39 IRQHandler .long Reserved40 IRQHandler .long DAC0 IRQHandler .long .long TSI0 IRQHandler .long MCG IRQHandler .long LPTMR0 IRQHandler .long Reserved45 IRQHandler .long PORTA IRQHandler PORTB IRQHandler .long

/* DMA channel 0 transfer complete*/ /* DMA channel 1 transfer complete*/ /* DMA channel 2 transfer complete*/ /* DMA channel 3 transfer complete*/ /* Reserved interrupt*/ /* Command complete and read collision*/ /* Low-voltage detect, low-voltage warning*/ /* Low leakage wakeup Unit*/ /* I2C0 interrupt*/ /* Reserved interrupt*/ /* SPI0 single interrupt vector for all sources*/ /* Reserved interrupt*/ /* UARTO status and error*/ /* Reserved interrupt*/ /* Reserved interrupt*/ /* ADC0 interrupt*/ /* CMP0 interrupt*/ /* TPM0 single interrupt vector for all sources*/ /* TPM1 single interrupt vector for all sources*/ /* Reserved interrupt*/ /* RTC alarm*/ /* RTC seconds*/ /* PIT interrupt*/ /* Reserved interrupt*/ /* Reserved interrupt*/ /* DAC0 interrupt*/ /* TSI0 interrupt*/ /* MCG interrupt*/ /* LPTMR0 interrupt*/ /* Reserved interrupt*/ /* PORTA Pin detect*/ /* PORTB Pin detect*/

Děkuji za pozornost...