

Vývojové prostředky

Richard Růžička

Vývojové prostředky

které se obvykle používají při vývoji aplikací pro MCU

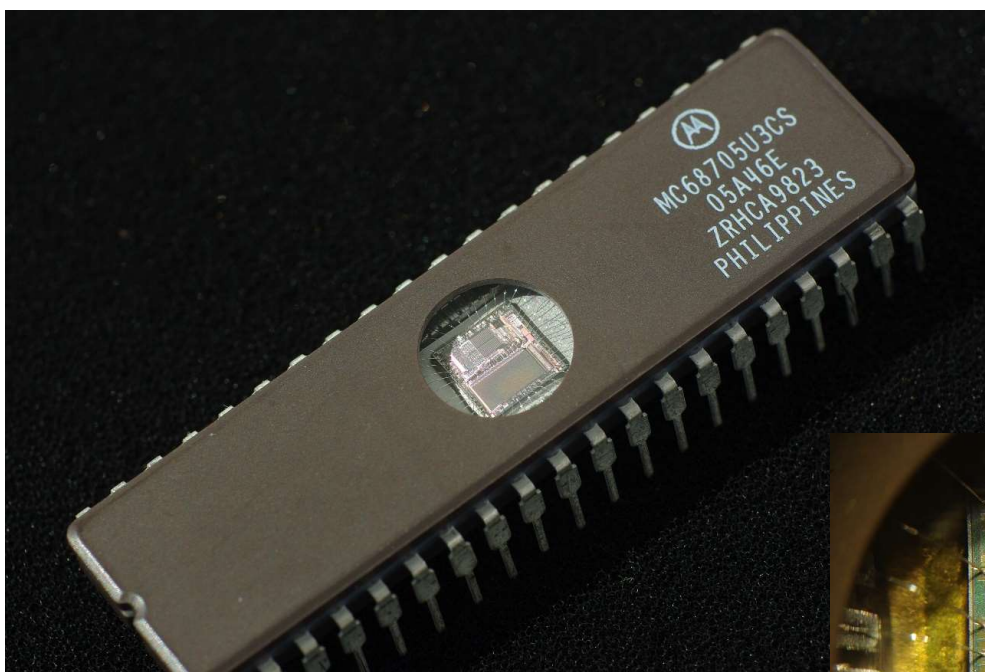
- Editor, dnes zpravidla součást IDE
- Překladač – assembler, C, ...
- Programátor nebo programovací/ladicí rozhraní
- Prostředky k ladění:
 - simulátor,
 - obvodový emulátor,
 - vývojová deska.

Programátor

= samostatné zařízení, do kterého se vloží MCU.

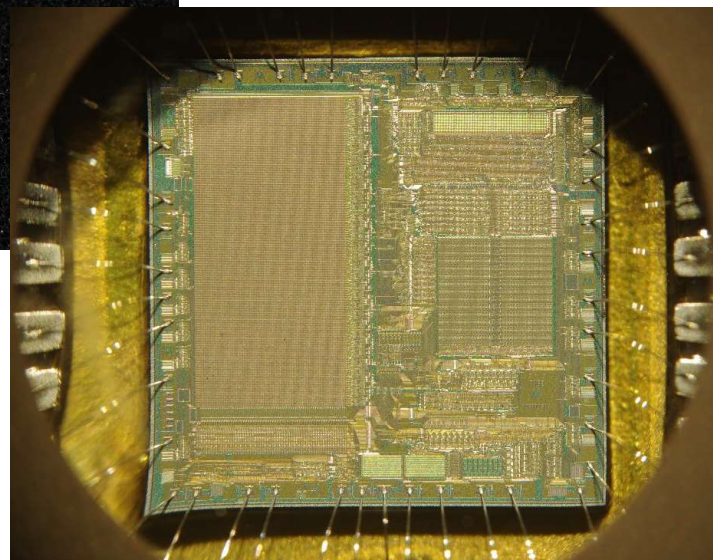
- Používá se patice s nulovou zakládací silou (ZIF), aby se minimalizovalo riziko poškození vývodů.
 - Zejména pro starší mikrokontroléry s PROM a EPROM.
 - ... mazací zařízení na EPROM...
 - Verze OTP – EPROM, ale pouzdro bez okénka!
-
- Pro velké série MCU s ROM (program vložen při výrobě čipu - maska).

MCU s EPROM pamětí



Okénko slouží ke smazání (vybití náboje z plovoucích hradel tranzistorů pam. buněk) pomocí UV záření.

Drahé pouzdro, většinou keramické s okénkem z křemičitého skla. Vzhledem k nutnosti hermetického zapouzdření čipu při širokém rozsahu teplot nepřípadá v úvahu levnější plast.

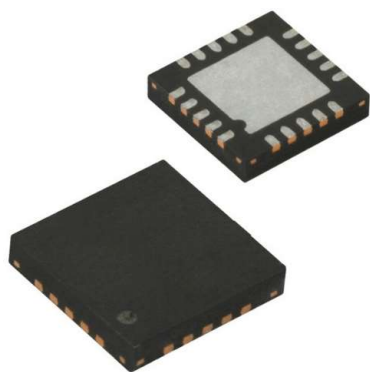


Programátor



Nezbytné pro čipy programované paralelně.
(pro programování se využívá většina pinů)

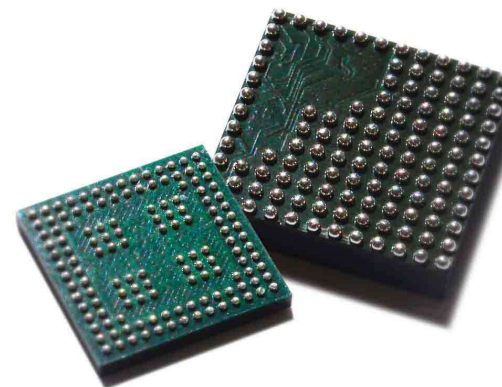
Současná pouzdra MCU



QFN
(Quad Flat No-lead)



LQFP
(Low-Profile Quad Flat Package)



BGA
(Ball Grid Array)

U současných pouzder je obtížné umístit je do patice (založit do programátoru). Vlastně se s tím ani nepočítá.

Počítá se s naprogramováním vnitřní paměti programu pomocí tzv. ISP, tedy přímo již osazeného (zapájeného) MCU v aplikaci (na desce).

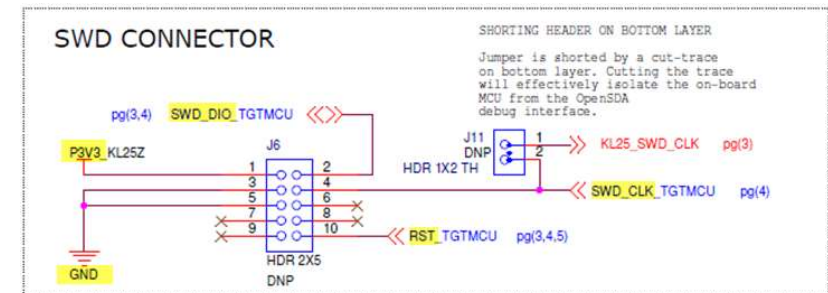
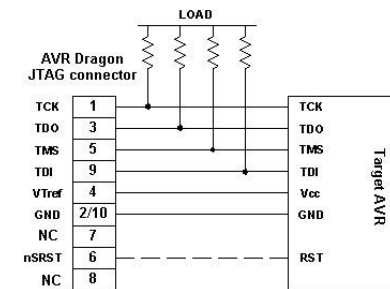
ISP

- In-System Programming - moderní přístup
 - vhodné pro MCU s Flash,
 - nabízí dnes každý výrobce,
 - umožňuje snadno upgradovat software i v zapájeném MCU,
 - většinou formou asynchronní sériové linky,
 - přepnutí do režimu ISP se děje kombinací správných napětí na pinech při resetu.

ISP

- Různé standardy vestavěného ladicího rozhraní

- JTAG
- MiniMON
- BDM
- SWD



- „Bootloader“ pro některé rozhraní MCU

- SPI, SCI, USB, ...

Ladicí rozhraní vs. bootloader

Bootloader

- je zpravidla **program (software)** v MCU, který
- očekává data na některém **běžném rozhraní** v definovaném formátu,
- přijatá **data uloží do paměti programu** (FLASH) a spustí jako program.

Ladicí rozhraní

- je zpravidla **realizováno hardwarově** (běží na pozadí celého MCU),
- umožní také nahrát program do FLASH, ale
- **má daleko více funkcí** – spouštění, trasování programu, čtení a zápis paměťových buněk a registrů, zachycování stavu sběrnice, breakpointy, ...

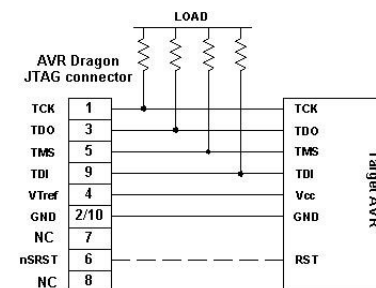
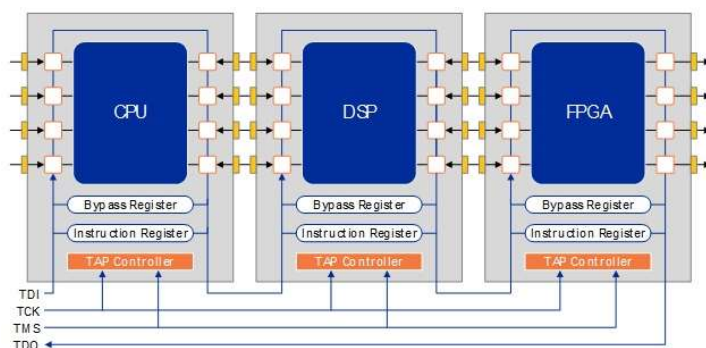
Ladicí rozhraní

- Pro připojení k PC je třeba mít vhodné zařízení.
- Dnes se připojuje zpravidla přes USB.
- Existuje řada výrobců, kteří se specializují na výrobu zařízení pro ladicí rozhraní (PE Micro, Segger, ...)

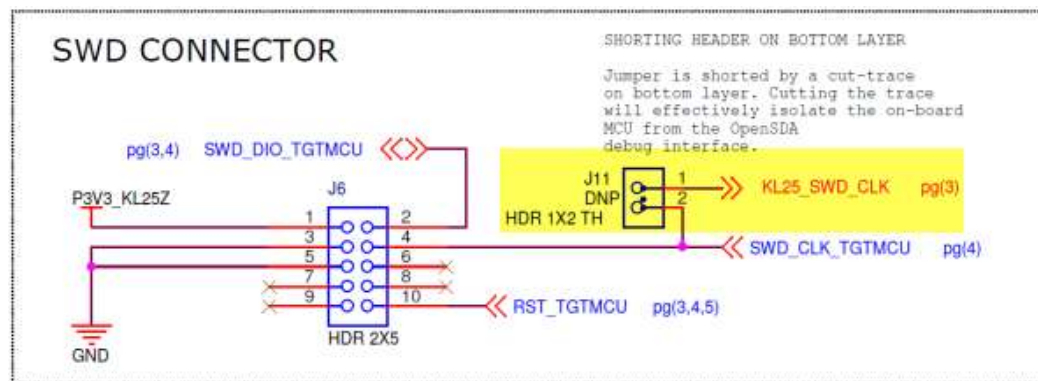


Standardy ladicího rozhraní

Často se používá JTAG

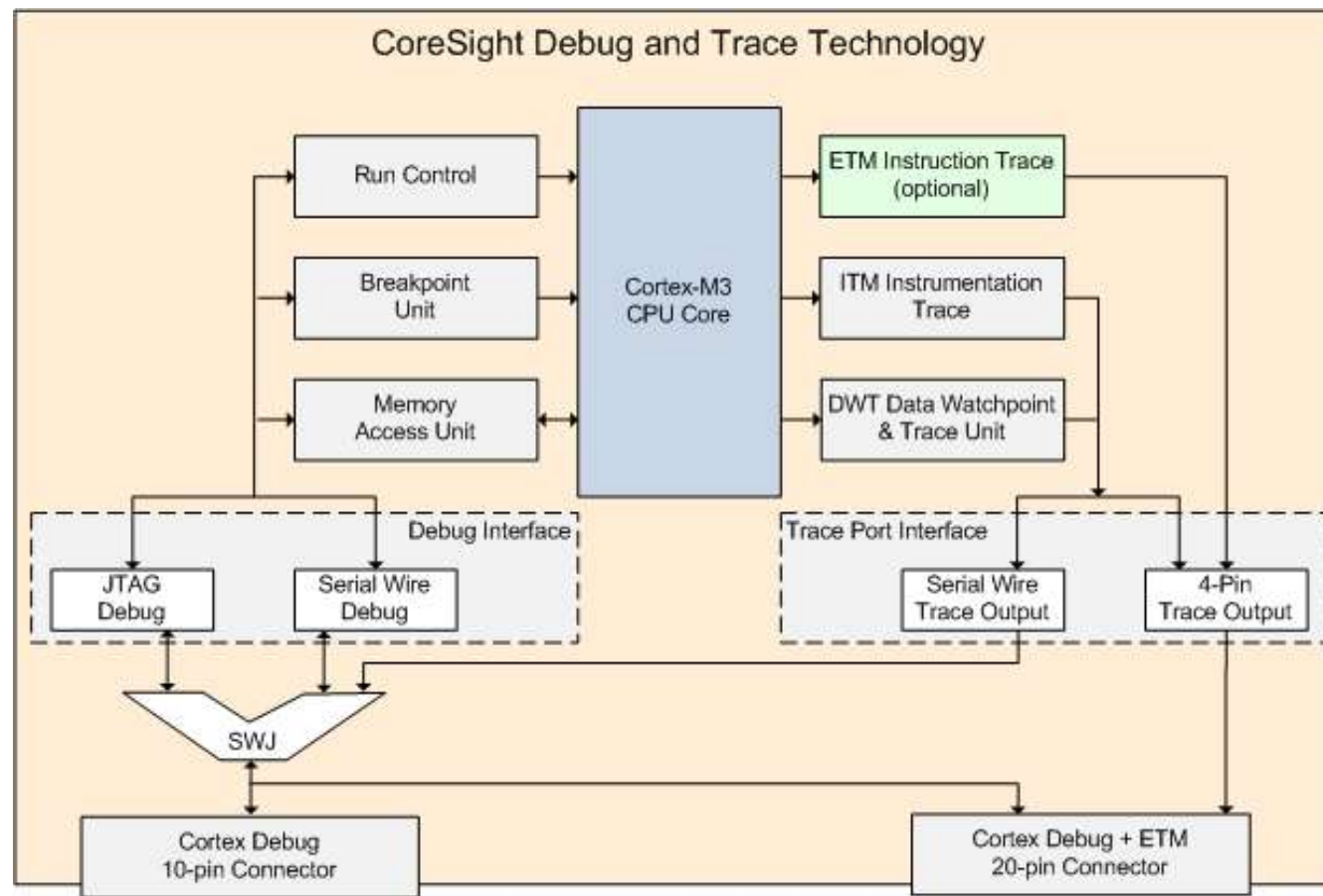


Řada výrobců má svoje řešení



Možnosti ladění moderního MCU

ARM CoreSight – komfortní debugger na čipu



Debug - frstFRDM/source/main.c - Kinetis Design Studio - C:\Users\ruzicka\workspace.kds

File Edit Source Refactor Navigate Search Project Run Processor Expert Window Help



Quick Access C/C++ Debug

Debug

<terminated> frstFRDM Debug [GDB PEMicro Interface Debugging]

<terminated, exit value: 0> C:\Freescale\KDS_v3\eclipse\plugins\com.pemicro.debug.gdbjtag.pne_2.3.6.2016022

<terminated, exit value: 0> arm-none-eabi-gdb

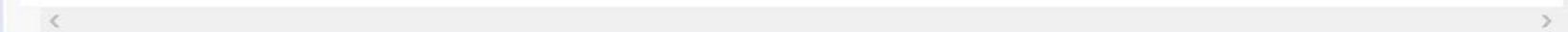


startup_MKL27Z644.S main.c fsl_gpio.h (gdb[9].proc[42000].threadGroup[i1],gdb[9].proc[42000].OSthread[1].thread[1].frame[0]

```
delay();
}

for(b=0;b<16;b++) {
    GPIO_ClearPinsOutput(GPIOA, 1<<13u); /* blue led on */
    delay();
    GPIO_SetPinsOutput(GPIOA, 1<<13u); /* blue led off */
    delay();
}

__asm("NOP"); /* something to use as a breakpoint stop while looping */
}
```



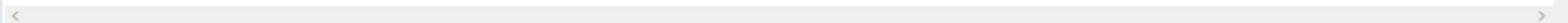
Console Tasks Problems Executables Memory

<terminated> frstFRDM Debug [GDB PEMicro Interface Debugging] arm-none-eabi-gdb

Note: automatically using hardware breakpoints for read-only addresses.

Temporary breakpoint 2, main () at ../source/main.c:53

53 BOARD_InitPins();



Breakpoints

Registers

EmbSys Registers

Peripherals

Modules

?o main.c [line: 80] [condition: b==5]

No details to display for the current selection.



Outline

Disassembly

Enter location he

```
.....
00000d58: bl 0xcbc <delay>
00000d5c: nop ; (mov r8, r8)
00000d5e: b.n 0xcf0 <main+16>
00000d60: orr.w r0, r0, #2399141888
00000d64: and.w r0, r0, #2399141888
DbgConsole_Printf:
00000d68: push {r0, r1, r2, r3}
00000d6a: push {r7, lr}
00000d6c: sub sp, #8
00000d6e: add r7, sp, #0
00000d70: ldr r3, [pc, #521]
```



SW nebo HW breakpoint?

```
0x100: 37 88 LDA @0x88  
0x102: 27 16 MULA #0x16  
0x104: 17 90 STA @0x90
```



```
0x100: 37 88 LDA @0x88  
0x102: FF FF BKPT  
0x104: 17 90 STA @0x90
```

Hardware Debug Trigger:

Condition: AddrMatch
Addr: 0x8002

```
0x8000: 37 88 LDA @0x88  
0x8002: 27 16 MULA #0x16  
0x8004: 17 90 STA @0x90
```

```
40 for(i=0;i<4;i++) {  
41     LED1_Neg();  
42     LED2_Neg();  
43 }
```

- Toggle Breakpoint
- Disable Breakpoint
- Breakpoint Properties...

Breakpoint Types

- Enable breakpoint in current context
- Enable breakpoint in all contexts

C/C++ Auto Breakpoints

C/C++ Hardware Breakpoints

C/C++ Software Breakpoints

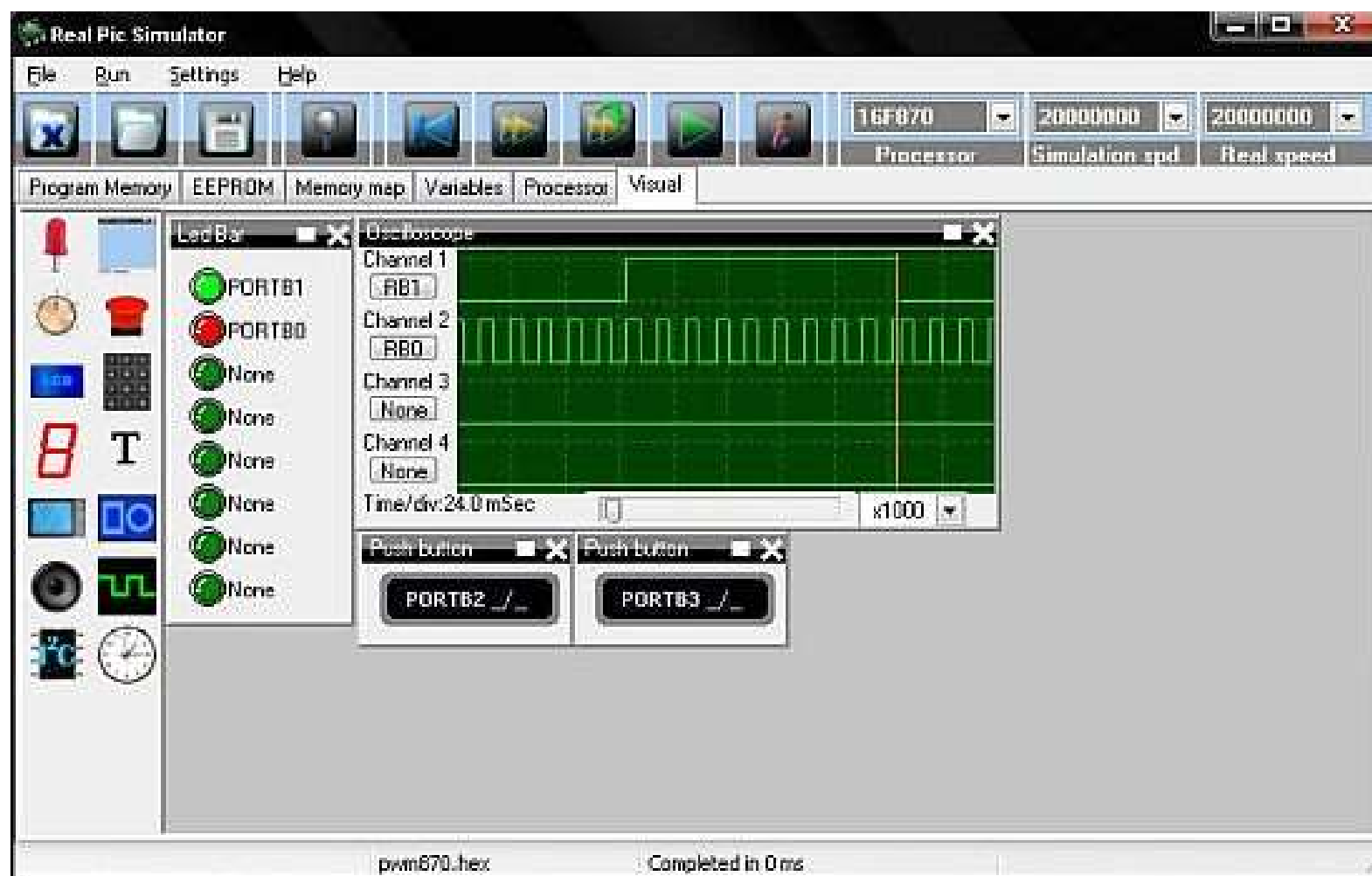
Simulátor

- Program, který běží v PC,
- v paměti udržuje model MCU,
- interpretuje program pro MCU,
- zobrazuje stav MCU.

Simulátor – výhody a nevýhody

- + Lze ladit kdekoli,
- + není potřeba mít MCU,
- + nehrozí poškození systému v případě fatální chyby,
- + není problém zobrazit veškeré informace o stavu MCU.
- neběží zcela v reálném čase,
- problematická simulace periférií, zejména neobvyklých,
- nemusí být zcela věrné.

Příklad: simulátor obvodů PIC (Microchip)



Simulátory dnes

Protože

- vestavěné ladicí prostředky v současných MCU jsou velmi komfortní,
- současné MCU jsou velmi složité,

Přestává se vyplácet tvořit simulátory.

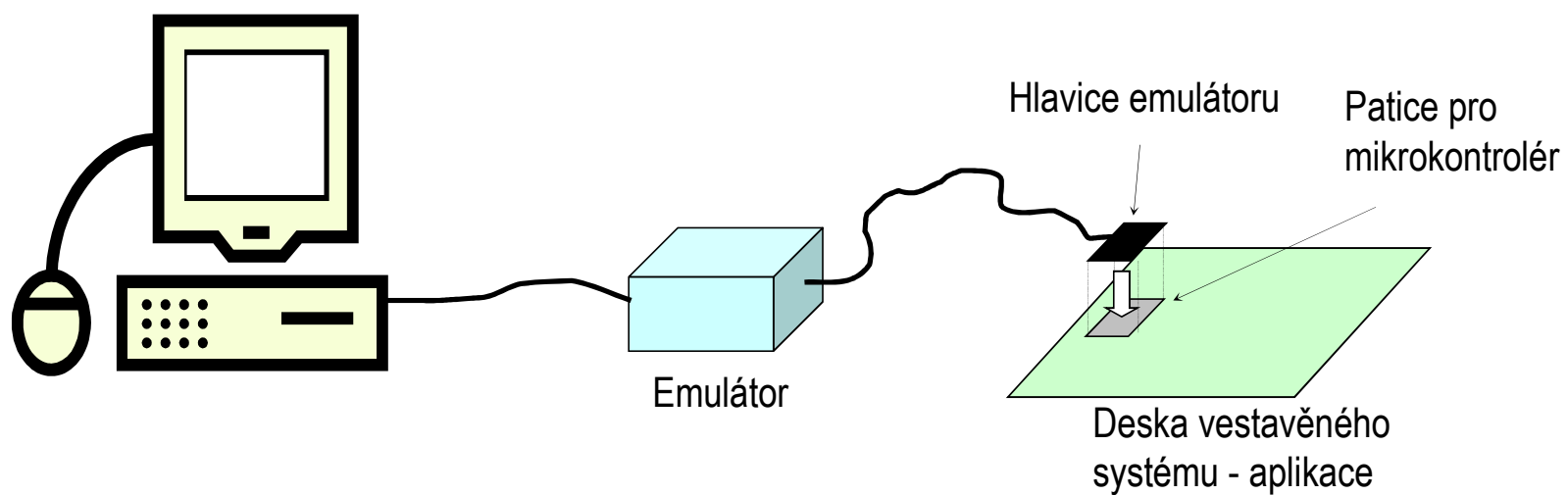
Dnešní MCU je za pár \$ a komfort ladění se vyrovnává simulátoru.

Například simulátor MCU Kinetis budeme dnes hledat těžko.

Obvodový emulátor

- Hardware, které se navenek chová jako MCU, ale...
- uvnitř kromě obvodů zajišťujících všechny funkce MCU jsou také obvody pro zobrazení stavu MCU a jeho ovládání (breakpointy, změny stavu atd.)
- končí hlavicí, která se zasunuje do patice v zařízení místo mikrokontroléru,
- je připojen k PC, kde běží aplikace podobná simulátoru - debugger.

Obvodový emulátor



Emulátor – výhody a nevýhody

- + Přesná emulace v reálném čase,
- + možnost komfortního ladění přímo cílové aplikace včetně skutečných nestandardních periférií.
- drahý prostředek, vyplatí se jen při opakovaném komerčním vývoji vestavěných systémů,
- vazba na pracoviště – obtížněji přenosný,
- zakoupení vede k nutnosti vyvíjet aplikace jen pro daný typ mikrokontroléru.

Emulátor HC08 – skut. provedení



Emulátory dnes

Platí prakticky totéž, co pro simulátory – komfortní prostředky k ladění, které dříve byly v emulátoru, jsou dnes k dispozici v každém MCU.

Proto budeme i emulátor MCU Kinetis hledat velmi těžko.

Vývojová deska

(development kit)

- Deska s mikrokontrolérem v základním zapojení a základními periferiemi.
- Slouží zejména k seznámení s MCU a ladění aplikací s běžnými periferiemi.
- Možnosti ladění závisí na možnostech ladění vestavěných v MCU.
- Výhodou je ISP, někdy bývá v MCU připraven tzv. monitor.

Jednoduchá a levná vývojová deska

Slouží pro základní seznámení se s MCU, demonstraci triviální aplikace, je na ní základní zapojení MCU s připojeným ladicím rozhraním. Bývají vyvedeny piny na konektor/plošky, často ve standardizovaném formátu.

Příklad pro Kinetis:
FRDM
(Freedom Board)



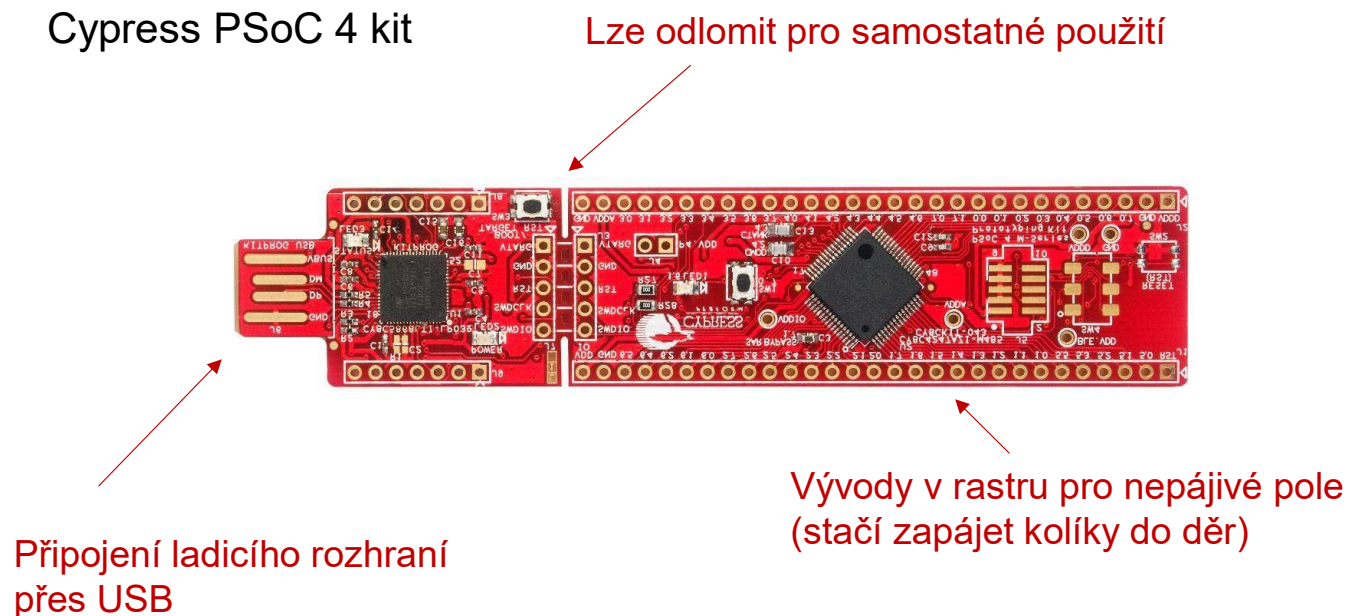
Připojení ladicího rozhraní
přes USB

Konektory pro „shield“ typu Arduino

Jednoduchá a levná vývojová deska

Slouží pro základní seznámení se s MCU, demonstraci triviální aplikace,
je na ní základní zapojení MCU s připojeným ladicím rozhraním.
Bývají vyvedeny piny na konektor/plošky, často ve standardizovaném formátu.

Příklad:
Cypress PSoC 4 kit



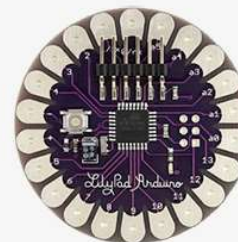
Jednoduchá a levná vývojová deska

Slouží pro základní seznámení se s MCU, demonstraci triviální aplikace, je na ní základní zapojení MCU s připojeným ladicím rozhraním. Bývají vyvedeny piny na konektor/plošky, často ve standardizovaném formátu.

Příklad: Arduino



Arduino Lilypad



Arduino Nano



Tower system pro Kinetis

Modulární systém vývojových desek, umožňující vytvořit a ladit snadno i složité aplikace.



Vývojová deska – výhody a nevýhody

- + Relativně levné ladění na skutečném mikrokontroléru,
- + ideální pro seznámení s MCU a výuku,
- + hotové zapojení MCU na desce (emulátor bez desky sám nefunguje).
- Použitelné jen pro aplikace se standardním zapojením a periferiemi,
- možnosti ladění odkázány na vestavěné prostředky v MCU,
- některé prostředky mohou být blokovány pro ladění