FAKULTA
ELEKTROTECHNICKÁ
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

RICE

Regionální inovační centrum elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Obecné zásady tvorby software

Mikroprocesorové řízení pohonů 2

Josef Štengl

www.rice.zcu.cz

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj pro inovace

OSNOVA PŘEDNÁŠKY

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Způsoby programování

Způsoby programováníz

Existuje více způsobů jak psát kód a jazyky, pro které je daný způsob programování určený nebo možný.

Základní způsoby programování jsou:

- deklarativní
- imperativní

Zjednodušeně lze rozdíly shrnout, že imperativní programy zápisem kódky krok za krokem se dosáhujeme žádané funkcionality, zatímco deklarativní jazyky specifikují cíl a algoritmizace je ponechána programu (interpretu) daného jazyka.

Deklarativní programování využívá rekurzivní volání místo smyček (while, for). Pro řídící systémy je rekurzivní volání funkcí nežádoucí z důvodu vyšší náročnosti na paměť a při programátorské chybě k možnému přetečení zásobníku a tím i možnosti nespecifikovanému běhu programu.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Způsoby programování

Deklarativní programování

Styl vytváření struktury a prvků počítačového programu vyjadřující logiku výpočtu bez popisu jeho řídícího prvku.
Pokouší se o minimalizaci implementace algoritmů v explicitních krocích.
Pro deklarativní programování je typické vyvarování se změny stavů a proměnných.
Jazyky pro deklarativní způsob psaní bývají turingovsky neúplné.
Zjednodušeně řečeno říkáte co chcete, ale ne nezbytně jak toho dosáhnout.
Deklarativní styl je nevhodný pro vytváření řídících algoritmůi i

Příklady jazyků pro deklarativní programování:
- SGML jazyky (HTML a podobné)
- regulární výrazy
- SQL

Deklarativním způsobem programování se nebudeme dále zabývat.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Způsoby programování

Imperativní programování

Způsob programování blízký člověku.
Kompilátoru jazyka píšete krok po kroku to co chcete.
Závisí na stavech programu (stavová mašina) kdy požadujeme od výpočetního systému přesně definované operace.
Kód je závislý na proměnných a změny jejich hodnot.
Poskytuje programátorovi široké možnosti algoritmizace, které přesně potřebuje.

Využívá tří základní skupiny příkazů
- přiřazení
  umožnuje uložit výsledek operace pro pozdější použití
- cykly
  umožňuje spustit sekvenci kódu spustit několikrát za sebou (for, while)
- větvení
  dovolí provést určitou část příkazu při splnění daných podmínek (if)

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Procedurální programování

Jedná se o způsob programování založen na principu (opakovatelného) volání procedur (rutin).

Procedury obsahují sérii výpočetních kroků včetně volání dalších procedur

Umožňují vytvářet logický celek programu zabývající se izolovaným problémem (moduly, modulární programování)

Umožňuje poměrně jednoduše a výpočetně efektivně řešit algoritmy cílové aplikace přirozeným způsobem, tak jak jsou zadávány při analýze problému.

Jeden z nejvýznamnějších procedurálních programovacích jazyků je jazyk C.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Objektově orientované programování

Je založen na principu modelování, kde si řešený objekt drží jeho vnitřní stavy; výkonný kód je přidružen k datům.
Umožňuje snadnější přenos kódu mezi různými projekty (abstrakce a zapouzdření)
Kód programu je založen na interakcích (vztazích) mezi těmito objekty.

Oproti čistě procedurálnímu programování má nevýhodu v relativní složitosti návrhu objektů, složitosti porozumění vzorů řešení a obtížnější řešení stavů aplikace. Dále mají větší spotřebu hardwarových prostředků (paměť, výkon CPU). Z těchto důvodů je OOP princip méně vhodný pro aplikace které jsou (dodnes) limitované výkonem hardware a náročné na časování a datovou propustnost, například nízkoúrovňové řízení pohonů.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Obecné zásady tvorby software

Proces návrhu je sled kroků, který umožňuje návrháři popsat všechny aspekty softwaru pro jeho úspěšný a efektivní návrh. Kreativní dovednosti, minulé zkušenosti, smysl pro to, co dělá „dobrý“ software a celková snaha o kvalitu jsou příklady zásadních faktorů úspěchu konkurenceschopného designu. Je však důležité si uvědomit, že proces navrhování není vždy přímým postupem. Začíná to představováním celku řešeného problému a jeho postupnému upřesňování, aby poskytla všechna důležitá vodítka pro konstrukci každého detailu. Základní principy návrhu umožňují softwarovému inženýrovi procházet procesem návrhu.

Vyvynulo se několik sad základních pravidel pro návrh konstrukce software. Jedním ze základních bodů jsou:

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Obecné zásady tvorby software

Abstrakce

Jedná se proces k získání co nejmenšího množství nutných informací, které jsou nutné pro pochopení procesu či konkrétního jevu relevantní pro konkrétní účel. Jedná se o krok k získání představy o základní funkcionality bez nutnosti znát podrobnosti pozadí problému či jevu.

Upřesnění (refinement)

Proces hierarchického zpracování rozkladem řešeného problému směrem seshora dolů, od nejvíce zobecněného až dokud se nedosáhne příkazů programovacího jazyka. V každém kroku je problém rozložen na podrobnější (konkrétnější) instrukce.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Obecné zásady tvorby software

Abstrakce a Upřesnění jsou doplňkové koncepty. Jejich užití je vhodné pro získání představy uživatelského rozhraní. Například pro regulaci k získání představy o možnostech a způsobu ovládání výsledného produktu, rychlosti zpracování ovládán a respektive minimální nutné rychlosti získávání reálných informací ze systému například proudu či napětích. Pro řízení pohonů bývají někdy velmi omezené dostupné komunikační prostředky s řídícím systém a je třeba si předem rozmyslet jaká a kolik dat je možné zadávat do řídícího systému, nebo je získat.

Příklad:

Pro získání hodnot napětí a proudů je lepší využít DAC jednotek, pokud jsou k dispozici. Ostatní komunikační kanály jsou mohou být velmi omezené rychlostí přenosů a získat data v reálném čase může být nemožné.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Obecné zásady tvorby software

Modularita

Rozdělení projektu na moduly. Každý modul představuje logický nebo funkční celek části projektu (například regulátor, čtení ADC, řízení PWM a podobně)

Architektura

Celková struktura software a organizace jeho součástí. Jedná se například o

- externí použité knihovny nebo moduly k použitému hardware
- rozdělení kódu do adresářů
- rozvržení struktury pro testování a verzí software.
- rozvržení struktury dokumentace jak daného software tak případně dokumentace hardware nutné k napsání software.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Obecné zásady tvorby software

Členění (struktura) programu

Strukturu programu lze rozdělit na horizontální a vertikální.

- Horizontální struktura definuje oddělené větve modulární hierarchie pro každou hlavní programovou funkci. To může být vyjádřeno jako samostatným souborem (adresářem) v adresářové struktuře zdrojového kódu.
- Vertikální struktura rozdělení naznačuje, že kontrola a práce by měly být distribuovány ze shora dolů ve struktuře programu. To je obvykle představeno podadresáři.

K jednotlivému modulu by se mělo programově přistupovat jen ve vodorovném směru!

Struktura dat

Jakým způsobem budou data reprezentována a organizována mezi moduly. Jedná se o vstupy a výstupy jednotlivých modulů, popřípadě sdílená data mezi proměnnými.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Obecné zásady tvorby software

Procedury software

Pokud je to možné je dobrým zvykem každou proceduru aplikace mít co nejvíce nezávislé na ostatních modulech.

Skrývání informací

Moduly by měly být navrženy tak, aby informace obsažené v modulu byly nepřístupné pro ostatní moduly, které takové informace nepotřebují.

Často je z důvodu optimalizace a snížení doby zpracování (CPU zátěže) kopírováním dat mezi moduly vhodné, mít jednu strukturu dat mezi více moduly. Je velmi vhodné zachovat minimální množství informací v takové struktuře.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Na co se zaměřit při návrhu software

Při návrhu software je nutno vzít na zřetel mnoho informací, které mohou mít zásadní vliv na výslednou velikost software. Jejich důležitost by měla zohledňovat cíle a očekávání kladené na výsledný produkt a tím i nároky na finanční, hardwarové a lidské zdroje.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Na co se zaměřit při návrhu software

Kompatiblita

Kompatibilita zahrnuje široké množství vlastností programu, mimo jiné:

- zpětná kompatibilita programu - jak je možno nahradit starší program novějším bez nutných nových nastavení nebo změn v uživatelském prostoru
- HW kompatibilita - schopnost programu pracovat na různých platfromách bez nutných dodatečných změn (endianita, šířka datových typů s pevnou a plovoucí řádovou čárkou, použité HW akcelerátory, kompilátory a podobně.

Rozšiřitelnost

Množství zdrojů, které je nutno alokovat ke stávajícímu software pro vytváření lze přidat nových funkcí programu. Jinak řečeno, jak je navržena základní architektury. Zde je nutno správně definovat rozsah použití software aby bylo možno účelně navrhnout jeho architekturu bez zbytečně velké robustnosti.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Na co se zaměřit při návrhu software

Modularita

Software by se měl skládat z co nejvíce dobře definovaných nezávislých komponentů.

Dodržení této zásady vede ke značnému zjednodušení údržby a možnosti izolovaného testování před vlastní integrací do výsledného software, nebo ke snadné využitelnosti v jiných projektech.

Mimo jiné to umožňuje i snadnější rozdělení vývoje mezi jednotlivé vývojáře.

Tolerance k chybám

Schopnost zotavení nebo oprav po selhání komponenty, zejména při chybách komponenty hardware (ADC převody, PWM a podobně).

Tento cíl je při návrzích řídících systémů pro řízení motorů opomíjen, jelikož je málo způsobů, jak se z problému zotavti. Při analýze (před vlastním psaní kódu) je velmi vhodné vědět, jak se systém má chovat v případě poruchy a jestli a jak je možno se z poruchy pokusit zotavit. Jedná se o zásadní prvek pro achitekturu software a proto by neměl být opomíjen.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Na co se zaměřit při návrhu software

Udržovatelnost softwaru

Schopnost jak snadno lze provést opravy chyb software nebo přidávání nových funkcí a komponent.

Dobře udržitelný software může být dán návrhem modulární struktury a snadné rozšiřitelnosti.

Zahrnuje i oddělování logických a funkčních celků do co nejvíce nezávislých bloků (modulů).

Spolehlivost

Množství zdrojů, které je nutno alokovat ke stávajícímu software pro vytváření lze přidat nových funkcí programu. Jinak řečeno, jak je navržena základní architektury. Zde je nutno správně definovat rozsah použití software aby bylo možno účelně navrhnout jeho architekturu bez zbytečně velké robustnosti.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Na co se zaměřit při návrhu software

Robustnost

Schopnost software vypořádat se vstupy mimo běžný pracovní rozsah zařízení nebo při hraničních podmínkách (stresu) jako jsou teplota, vlhkost, elektromagnetické rušení a podobně.

Dále je schopnost se vypořádat s chybami harware (nebo jestli a jak jsou zpracovny chybové stavy použitých periférií), a chybami vlastního harware (například jestli existuje parita pro kontrolu paměti)

Zabezpečení

Do jaké míry má software mít odolnost proti nepřátelským vlivům.

Použitelnost

Uživatelské rozhraní mezi člověkem a strojem.

Komfort obsluhy a pro jaké lidi (vzdělání, věk a podobně) je výsledný produkt cílen.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Na co se zaměřit při návrhu software

Výkon

Rychlost měření a zpracování dat

- rychlost řídící smyčky
- doba odezvy systému
- množství přenášených data do nebo vně systému.

Přenositelnost

Do jaké míry má být software nezávislý na mikroprocesorové platformě, na které běží.

Škálovatelnost

Schopnost software se dobře přizpůsobovat rostoucímu množství dat nebo přidaným funkcím či počtu uživatelů.

S problémem škálovatelnosti se u mikroprocesorových řídících systémů v naprosté většině případů nesetkáme, protože řídí systém je většinou jedna aplikace běžící na jedné hardwarové platformě.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Čemu se při návrhu SW vyhnout

- Proces vývoje SW by se měl vyvarovat „tunelovému vidění“

  Dobrý návrhář by měl zvážit i alternativní přístupy (a studovat je), přičemž se každý musí posoudit na základě požadavků a dostupných zdrojů, které má k dispozici. Návrh by měl být navázán na analytický model.
- Z návrhu by mělo být zřejmé, kterou část analýzy zpracovává

  Vzhledem k tomu, že jeden prvek návrhového modelu lze často vysledovat zpět k více požadavkům, je nutné mít k dispozici prostředky pro sledování toho, jak byly návrhovým modelem splněny požadavky.
- Proces vývoje SW by se měl vyvarovat „vynalézání kola“

  Systémy jsou konstruovány pomocí sady návrhových vzorů, z nichž mnohé se již dříve pravděpodobně objevily. Tyto vzory by měly být vždy zvoleny jako alternativa k novému objevu. Čas je neúprosný a zdroje jsou omezen, a čas na design by měl být investován do skutečně nových nápadů a pokud je to možné tak integrovat vzory, které již existují.
- Návrh by měl minimalizovat intelektuální náročnost mezi vlastním softwarem a problémem, který řeší v reálném světě

  Struktura softwarového řešení by měla, pokud je to možné, napodobovat strukturu řešeného problému.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Na co se zaměřit při návrhu software

- Návrh by měl dbát na jednotnost a integraci

  Před zahájením projektu by měla být definována pravidla stylu a formátu pro návrhářský tým. Integrace znamená, že by měli být navržena rozhraní mezi jednotlivými komponentami návrhu.
- Návrh by měl mít strukturu umožňující jednoduché změny
- Návrh není kódování, kódování není návrh

  I když jsou pro komponenty programu vytvořeny podrobné procedurální návrhy, úroveň abstrakce návrhového modelu je vyšší než zdrojový kód. Jediné rozhodnutí o návrhu učiněné na úrovni kódování by mělo řešit drobné detaily implementace, procedurálního návrhu.
- Revize návrhu

  Návrh by měl být přezkoumán (nejlépe jinou zkušenou osobou), aby se minimalizovaly koncepční (sémantické) chyby. Při kontrole návrhu je někdy tendence zaměřit se na detaily a uniká podstata celku. Vývojový tým by měl zajistit, aby byly řešeny hlavní koncepční prvky návrhu (opomenutí, nejednoznačnost, nekonzistence), než začnou vytvářet vlastní implementaci.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Přístup k perifériím hardware

Řídící aplikace na mikroprocesoru je ve většině případů jedinou aplikací, která na daném hardware běží a použití operačního systému je v tomto případě nežádoucí s ohledem na vyšší výpočetní výkon a možnou vyšší latenci.

Pokud není použit operační systém, je třeba rozhodnout jakým způsobem přístupu bude probíhat komunikace ke konkrétním perifériím. Způsob přístupu je zásadní pro návrch struktury software.

Naopak, služby operačního systému jsou přínosem v případě složitějších systémů, které používají náročnější komunikační periférie jako USB, Bluetooth nebo ICP/IP komunikaci.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Přístup k perifériím HW

Pro přístup k perifériím se používají 3 základní způsoby přístupu

- blokující (synchronní)
- událostní za pomoci přerušovacího systému procesoru
- za pomoci DMA kanálů

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Přístup k perifériím HW

Blokující přístup

K přístupu k periférii se obvykle využívá jedna nebo více funkcí, při rozlišování směru přístupu k periférii.

Vykonávání kódu je synchronní (sekvenční) s ohledem přístupů k datům

+ obslužná funkce má nejjednodušší přístup k datům periférie

+ pro jednoprocesorový systém nedochází k porušení konzistence dat, obslužná funkce čeká na kompletní dokončení operace

– obslužná rutina čeká na dokončení vstupně výstupních operací.

– z důvodu možných poruch při komunikace je velmi vhodné mít časovou kontrolu doby trvání přístupu k periférii (timeout)

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

Přístup k perifériím HW

Událostní přístup

Obslužná rutina přístupu k periférii je vykonávána v obsluze přerušení procesoru.

Asynchronní přístup k periférii a datům.

+ CPU zátěž je spotřebována v nezbytné nutné míře k přístupu k periférii plus čas nutný pro vstup a výstup z módu přerušení mikroprocesoru.

+ Umožňuje rychlou odezvu na události od periférie.

– je třeba zajistit konzistenci dat z důvodu asynchronního přístupu (atomické proměnné, časování), náročnější na desing software než v blokujícím režimu

– pro zachycení chybových stavů je řízení komunikace je třeba separátní rutiny

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

DMA přístup

Vhodný na opakované přenosy větších objemů dat (ADC, DAC, PWM, komunikace s daty o fixní velikosti paketu)

+ nejmenší časová náročnost na CPU

+ pro velmi rychlé komunikace to může být jediný způsob jak ji technicky na dané hardwarové platformě realizovat.

- pro malý objem dat může být více náročné na čas procesoru než předchozí případy

- nevhodné pro periférie s nestálou délkou dat

- nejsložitější obsluha (je třeba konfigurovat minimálně danou periférii a DMA)

- nevhodná pro přenosy malých objemů dat

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace

FAKULTA
ELEKTROTECHNICKÁ
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

RICE

Regionální inovační centrum elektrotechniky
Fakulta elektrotechnická
Západočeská univerzita v Plzni

Děkuji za pozornost!

Adresa:
Univerzitní 26
306 14 Plzeň
Česká republika

Tel: +420 377 634 152
Fax:

Email: jstengl@fel.zcu.cz

www.rice.zcu.cz

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI
OP Výzkum a vývoj pro inovace