

Semestrální projekt VI2 – Úterý 9:00

V LabVIEW vytvořte aplikaci, která bude splňovat následující body:

Proces číslo 1

- 1) Kontinuálně měřte **REÁLNÝ** signál pomocí měřicí karty PCI6221 obsažené v PC v EB206. Tento signál má následující parametry:

- Tvar: Trojúhelník
- Frekvenční pásmo: 10Hz až 25kHz
- Amplituda: dle karty
- Offset: dle karty

(Rady: Pamatujte na vzorkovací teorém při nastavování parametrů karty)

- 2) Naměřený signál zobrazte v grafu (jakýkoliv zobrazující **sadu** dat) s proměnlivou délkou vyhodnocovacího okna T_w . Délky půjdou nastavit v hodnotách **100 ms, 250 ms, 500 ms a kompletní okno tzn. 1s**. Na Y ose budou **Volty**, na X ose bude čas v **milisekundách**.

(Rady: Vzpomeňte si na to, co určuje délku okna, existuje vztah mezi vzorkovací frekvencí a vyčtenými vzorky. Pro „nastavovač“ užíjte ENUM či RING. Nedoporučuji používat Numeric control)

- 3) **Signál se bude jevit jako stacionární a bude tedy zatriggrován (nebude tzv. „ujíždět“)**. Signál bude **začínat průchodem nulou**. K překreslení grafu (*toto se děje pořád cyklicky samozřejmě, ale uživateli se graf, kromě šumu, jeví jako neměnný*) dojde až po dalším průchodu nulou. Pokuste se co nejvíce eliminovat zkreslení šumem. Pokuste se eliminovat překlápění signálu po půlperiodě.

(Rady: Zkreslení šumem lze eliminovat pomocí „znecitlivění“ detektoru (dělalo se na cviku). Překlápění řešte, až budete mít vyřešen šum. Je to jednodušší. Robustnost si testujte na generovaných datech (VI1 projekt generátor trojúhelníku či nějaká funkce + dostatečný šum))

- 4) Ze zatriggrovaneho signálu vypočtete hodnotu U_{RMS} (tentokrát můžete užít knihovních funkcí ☺), dbejte na to, aby tento výpočet byl proveden korektně **a nebyl zkreslen** (necelistvý počet period).

- 5) Jakmile dojde k překročení hodnoty U_{RMS} přes triggerovací hodnotu $U_{RMS_TRIGGER}$ (jiný trigger než ten výše) nastane událost, která musí vyhovovat následujícímu:

- a) K triggeru dojde pouze **JEDNOU**. K novému zatriggrování smí dojít pouze tehdy, když signál klesne opět pod hladinu $U_{RMS_TRIGGER}$ (jedná se tedy o hranový detektor citlivý na vzestupnou hranu)
- b) V okamžiku splnění podmínky odešlete vyhodnocovací okna **$T_w(n-1)$, $T_w(n)$ a $T_w(n+1)$** do **Procesu číslo 3**, kde budou tato data zpracována. (Tedy okno z minulé iterace, aktuální iterace s triggerem, a okno z následující iterace)

DŮLEŽITÉ: K předání dat NESMÍTE použít lokální/globální proměnné. Použijte vhodnou synchronizační techniku. Dbejte na to, aby nedošlo ke ztrátě žádných dat.

Proces číslo 2

- 1) Simultánně s výše uvedeným provádějte na **VŠECH** datech následující Signal processing:
 - a) **Převzorkujte signál na hodnotu $f_{vz} = 5\text{KS/s}$** – dbejte na **eliminaci aliasingu** pomocí vhodného filtru a metodiky
 - b) Vypočtete a správně zobrazte frekvenční spektrum pro převzorkovaný signál (5kS/s)
 - c) Vypočtete a správně zobrazte frekvenční spektrum pro originálně navzorkovaný signál
 - d) Vypočtete periodu a frekvenci signálu z průchodů nulou
 - e) Vypočtete periodu a frekvenci signálu z frekvenčních spekter
 - f) Vypočtete U_{RMS} hodnotu z převzorkovaného signálu $U_{\text{RMS}5\text{KS/s}}$ a zobrazte ji společně s U_{RMS} v jednom grafu typu CHART

(Rady: Z výše psaného vyplývá, že musíte spolu s měřenými daty předávat i hodnotu U_{RMS})

Proces číslo 3

- 1) V okamžiku triggeru uložte celou sadu dat (všechna tři T_W) do formátu „.tdms“. Název souboru bude ve tvaru „vaslogin_rok_mesic_den_hodiny_minuty_sekundy.tdms“. Čas uvedený v názvu data je právě ten, kdy proběhl trigger. (Neplatí tedy ten, kdy byl soubor ukládán).
- 2) Na panelu na stolech v laboratoři rozsvítte s triggerem hardwarovou LED diodu, která bude 1s svítit a poté zhasne.

Požadavky a nezmíněná doporučení

- Všechny procesy budou v jediném VI nazvaném „vaslogin_VI2projekt_main.vi“
- Vše si pořádně testujte na generovaných datech, přístroje doma nemáte
- Kód BUDE vhodně okomentován
- Kusy kódu vhodně zabalte do subVI, tak, aby se celkový kód vešel na jednu obrazovku fullHD monitoru. Odmítám tahat posuvníky ☺ **Nezapomeňte pak tyto subVI odevzdat!**
- Než použiji „Stacked Sequence“ či „Sequence“ tak se opravdu podívám, zda je to nutné a nedělá tuto funkci za mě princip dataflow sám o sobě. Snažte se použití této sekvence vyvarovat tím, že si vytvoříte subVI
- Nepoužívejte expresní VI a snažte se vše poskládat sami z elementárních VI, či exemplů. Jde hlavně o to, abyste pochopili souvislosti a principy, které se v předmětu učíme
- Zbytečný kód smažte, ať mne nemate. (Tzn. bulharské konstanty z vývoje, čarování s poli či matematikou, míchání času a frekvence atd.) V odevzdaném kódu tedy bude pouze to, co tam má být a plní to funkci, kterou má
- Snažte se vše naprogramovat co nejefektivněji a nejjednodušeji. Projekt je poměrně lehký. V ten moment jak budete čarovat s referencemi, property nodama, invoke nodama je něco asi špatně.
- Kód by neměl obsahovat moc konstant. Kde je to nutné, tam ano (nastavení měřicí karty), ale jinak výpočty (např.: počet vzorků pro délku okna)
- Snažte se kód tvořit úhledně a dle principů, které vám říkal pan docent. Tj. dataflow, zleva/doprava, co nejvíce rovné dráty, bez zbytečného křížení se

DOKUMENTACE VE WORD ČI PDF VÁM BUDIŽ ODPUŠTĚNA, BUDE PROVEDENA KOMENTÁŘI V KÓDU

DATUM ODEVZDÁNÍ: Neděle 14.4.2019, půlnoc z neděle na pondělí, na cvičení 16.4.2019 bude probíhat obhajoba projektu a diskuze bodů. Projekt odevzdejte na mail jan.baros@vsb.cz v zipovém či rarovém souboru s vaším loginem.

Případné dotazy na cvičení, či přes mail.