Hakkel Tamás, Halász Viktor Olivér
2015.02. 18. és 25.
1. csoport (szerda)

Jegyzőkönyv

A mérést végezték: Hakkel Tamás, Halász Viktor Olivér
A mérés ideje: 2015. február 18. és 25.
A mérés helye: PPKE ITK épülete, 320-as terem/mérőlabor

A mérés célja:

A LabView programrendszer megismerése, alapjainak elsajátítása

A mérendő objektumok:

Ø

A méréshez felhasznált műszerek:

LabView programrendszer

A mérés menete:

Az órán a kiadott feladatokat a LabView programkörnyezetben végeztük el, felhasználva előzőleges, illetve az órára való felkészüléssel megszerzett tudásunkat felhasználva. Amennyiben elakadtunk, a gyakorlatvezetők segítségét kértük, illetve az interneten és a Helpben kerestünk segédanyagot.

Az elvégzett feladatok:

1) Egy tetszés szerinti típusú kapcsolót bekapcsolva gyulladjon meg egy szögletes kék LED. A m/s mértékben beadott sebességet írja ki és mutassa meg egy tetszés szerinti formájú kijelző km/ó egységben.

   a. Minden egyes feladatot azzal kezdtük, hogy megjelenítettük a controls palette-t és tools palette-t (view menüből), továbbá az átláthatóság érdekében osztott képernyőben jelenítettük meg a block diagramot és a front panelt (window/tile…).

   b. Lefektettünk egy szögletes LED-et a listából, amelynek átállítottuk a színét kékre, majd a kiválasztottunk egy kapcsolót. Ezek után a block diagram ablakban összekötöttük a kapcsoló kimenetét a LED bemenetével. Ezek után futtattuk, a kapcsoló mind lekapcsolt, mind felkapcsolt állapotában, az utóbbi esetben a LED világított (a front panelben természetesen).

   c. Kiválasztottunk egy numerikus controlt és egy numerikus indikátort a listából. Ezután a control tulajdonságainál a display format fül alatt, a format string mezőnél átállítottuk a mértékegységet m/s-ra (ennek beírásakor fontos a „#_” a felesleges tizedes jegyek elhagyása végett). Ugyanezt elvégeztük a az indikátorra is, km/h-val. Ezután beillesztettünk a block diagramba a function menüből egy szorzás (multiply) műveletet, illetve egy valós numerikus állandót (DBL numeric constant), melynek értékét 3,6-ra állítottuk. Végül a megfelelő módon összekötöttük a numeric controlt és az állandót a szorzás művelet, amelyet pedig az

indikátorral.

2) Alakítsa át az 1. pont feladatát olyanra, hogy csak akkor történjen mértékegység átszámítás, ha a kapcsoló ki van kapcsolva, és a mértékegység átszámítás legyen subvi.

   a. Az első feladatot „szerkesztettük át". Az 1 b)-ben leírt összeköttetést egy case struktúrába húztuk (functions/structures/case structure), amelynek logikai értékét most true-ra állítottuk. Ezután az 1 a) kapcsolóját összekötöttük a case struktúrával. Így akármikor felkapcsoltuk a kapcsolót, a mértékegységátváltás megtörtént. (Természetesen nem 0 adatokat is írva a numerikus controlba.) A subvi megvalósításhoz a konnektor táblán kijelöltük azokat a mezőket, amelyik a mi bemenet-kimenet párosításunknak (egy darab bemenethez egy kimenet tartozik) megfelelő volt. Végül a block diagramban kijelöltük a 3,6-tal való szorzást, és SubVIt csináltunk belőle. (Természetesen annak mentéséről sem feledkeztünk meg.)

3) Egy nyomógombot egyszer megnyomva induljon el egy kockadobás és numerikus kijelzőn és mutatós műszeren jelezze ki a dobás értékét. Ha az eredmény 6-os akkor gyulladjon ki egy kör alakú zöld LED.

   a. Kiválasztottunk egy OK-gombot, amelynek címkéjét a feladathoz jobban illő „Dobás"-ra állítottuk. Az OK-gombot összekötöttük egy case struktúrával (true-ra állítva), ezután minden további elemet a case struktúrába helyeztünk. Egy random number generatort és egy 5 értékűre állított (long integer típusú) állandót összeszoroztunk, majd ezt, és egy 1 értékű (long int) állandót összeadtunk. Erre azért volt szükség, mert a véletlenszám-generátor 0 és 1 közötti értékeket ad vissza, nekünk viszont 1 és 6 közöttiekre van szükségünk.

   b. Annak érdekében, hogy egész számokat kapjunk eredményül, az eddigi értéket átkonvertáljuk byte integerre (ezen konvertáló elemmel kötjük össze az összeadást). Ezt az elemet kötjük össze az indikátorral, amely futtatáskor rendeltetésnek megfelelően működik.

   c. Végül megvizsgáltuk ennek az értéknek az egyenlőségét egy 6 értékű állandóval (összekötöttük a konvertáló elemet és a 6-állandót az equal? elemmel). Az egyenlőségvizsgáló elemet (comparison/equal?) pedig összekötöttük egy LED-del. Amennyiben 6-ost „dobtunk" (az indikátor szerint) a lámpa felizzott.

4) Mérje meg és jelezze ki a mértékegység átszámító subvi futási sebességét! Alakítsa át a 2. pont programjait úgy hogy azok pontosan 10000 szer fussanak le. Határozza meg az egyes programok egyszeri lefutása által felhasznált futási időt!

   a. A 2. feladatot átmásolva, és új néven lementve szerkesztettük át a 2. feladatot (ügyelve a SubVI-re). A feladat alapját a három darab parcellából (frame-ből) álló szekvencia képezi, így megfelelő sorrendben hajtódnak végre a műveletek.

   b. Az első parcellába illesztettünk egy Tick Count időmérő függvényt, amely az átváltás elindítása előtt elindít egy időmérést. A második parcellába egy For ciklust (loopot) helyeztünk el, amit egy konstans segítségével 10000-szer végzünk el (a loop countot összekötjük vele). A ciklusba raktuk bele az átváltást elvégző feladatrészeket. A harmadik parcellába belehelyeztünk egy újabb időmérőt (Tick countot), amelynek értékéből kivontuk

az első parcella időmérőjének értékét, majd ezt elosztottuk 10000-rel. A végeredményt ebben a parcellában az átváltás időtartamát jelző indikátorral kötöttük össze.

5) Az előlapon helyezzen el egy kapcsolót és egy numerikus kijelzőt. Mérje meg emberek időérzékelő képességének pontosságát. A feladat az, hogy lehetőleg egy másodpercig tartsa bekapcsolva a kapcsolót. A visszakapcsolás után írja ki a kijelző a tényleges időtartamot. Ha az eltérés 10%-ot nem halad meg, akkor gyújtson ki egy zöld LED-et.

   a. Ehhez a feladathoz először elhelyeztünk egy nyomógombot, egy numerikus indikátort és egy LED-et. A block diagramon újfent szekvenciát kellett használnunk. A szekvencián kívül hagytuk a nyomógombot, amelyről két lokális változót készítettünk (Create/Local variable). A szekvencia első és harmadik parcellájába egy-egy While loopot helyeztünk és beléjük raktuk a lokális változókat. Az első ciklus feltétele az, hogy akkor álljon le, ha a lokális változó értéke igaz lesz (azaz a szekvencia, és így a feladat, csak akkor halad tovább, ha lenyomjuk a gombot). A harmadik parcella loopjában a feltétel az, hogy addig ismétlődjön a ciklus -és addig ne haladjon tovább a feladat-, amíg a lokális változó értéke igaz (tehát amíg le van nyomva a gomb).

   b. Az időméréshez egy-egy tick countot helyeztünk a 2., illetve 4. parcellába, amelyek értékeit double típusra konvertáltunk, és kivontuk őket egymásból, ezt az értéket jelenítettük a tényleges idő kijelzőjén. Továbbá ebből az értékből kivontuk az egy másodpercnek megfelelő 1000-t (mivel jelenleg ms-ban mérünk), vettük az abszolút értékét, és ezt összehasonlítottuk az 1000 (ms) 0.1-szeresével. Ha az előbbi kisebb, mint az utóbbi, a LED felvillan.

6) Készítsen egy szinusz, háromszög, négyszög -generátorból és voltmérőkből álló műszer-együttest. Szabályozható legyen a generátor kimenő és offszet feszültsége valamint frekvenciája. Az egyik voltmérő mutassa a mért jel effektív értékét, a másik pedig a csúcsértékét.

   a. A feladathoz elhelyeztünk 3 numeric controllt, melyeken be lehet állítani a jelgenerátor frekvenciáját, amplitúdóját és offset feszültségét. A kimeneti értékek megjelenítéséhez pedig három waveform chart-ot tettünk fel a front panelre, illetve mindhárom alá két numeric indicatort, melyek a három jel effektív feszültségét és csúcsértékét jelenítik meg.

   b. Mindhárom generátor magja egy for ciklus. A szinusz jelgenerátor a következő kifejezés alapján küldi a chartra a jelet: $A * \sin\left(\frac{i}{N} * f * 2\pi\right) + o$, ahol A az amplitúdó, i a ciklusváltozó, N a ciklus maximumértéke, f a frekvencia és o az offset feszültség. Az $\frac{i}{N}$ tényező biztosítja, hogy egyenletesen küldi a függvény az értékeket a chartra, f-fel való szorzással vízszintes irányban sűrűsítjük a szinusz hullámot, hogy frekvenciája megfelelő legyen, A az y tengely mentén nyújtja a függvényünket, o-val pedig eltoljuk az y tengely mentén. A kifejezés műveleteit grafikusan vittük fel a block diagram ablakra. A ciklusmagokba 10 ms késleltetést illesztettünk be, mivel ennél sűrűbb mintavételezésre nincs szükség.

   c. A szinusz-jel generátortól csak kicsit különbözik a négyszögjel generátor kifejezése: $A * sign\left(\sin\left(\frac{i}{N} * f * 2\pi\right)\right) + o$.

d. A háromszög-jelet az előzőektől nagyban eltérő módszerrel állítottuk elő. A következő kifejezés értékeit ábrázoltuk charton: $A * abs\left[\frac{\left(i-\frac{N}{2}\right)}{\frac{N}{2}}\right] + o$. Mivel ezzel a kifejezéssel csak egy periódust tudunk leírni, a frekvenciát az előzőekben ismertetett módszerrel nem tudtuk beállítani, ehelyett a ciklusmagba illesztett késleltetés értékét állítottuk be a frekvenciának megfelelően: $\frac{1000\,ms}{N*f}$, így N lépés után pont az eltelt idő: $1s * \frac{1}{f}$, ami egyenlő a periódusidővel. Mivel eltérő a szinusz-négyszög jelgenerátor és a háromszög jelgenerátor ciklusmagjában levő késleltetés, ezért két párhuzamosan futó for ciklussal oldottuk meg a három jel generálását.

e. Az effektív érték kiszámításához az alábbi képletet használtuk: $\sqrt{\frac{1}{T}\int_0^T f^2(t)dt}$, ahol a t helyére a ciklusváltozónk értékét írtuk. Az integrál értékét a Riemann integrál definíciója alapján becsültük a következőképp: $\int_0^T f^2(t)dt \approx \sum_{t=0}^{T} f^2(t)$

f. A csúcsértéket egyszerű maximumkeresés-algoritmussal oldottuk meg: folyamatosan vizsgáljuk, hogy az aktuális érték nagyobb-e, mint az eddig legnagyobbként eltárolt érték, és ha igen, akkor lecseréljük a legnagyobb értéket.