

LabVIEW 1

Kékesi Kristóf
NEPTUN kód: ZI6I4M
Mérőpár: Bor Gergő

Mérés ideje: 2024.03.13. 15:15-18:00

Mérés helye: Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar
1083 Budapest, Práter utca 50/A 321-es labor
kekesi.kristof.mihaly@hallgato.ppke.hu

Kivonat—

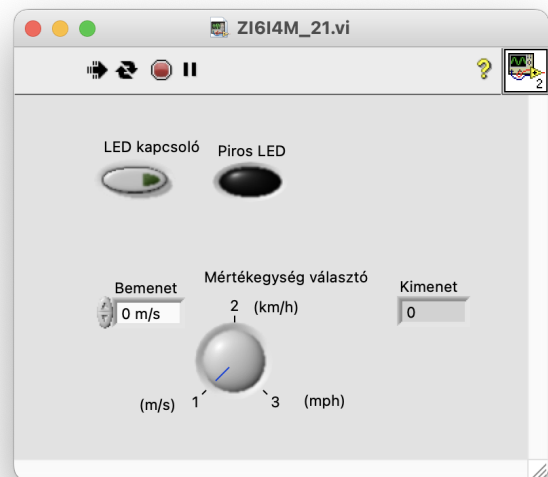
Az a jegyzőkönyv leírja a március 13-án megoldandó LabVIEW feladatok megoldásának menetét és a reprodukálásához szükséges információkat.

Ezek a feladatok röviden: m/s-ről km/h-ra váltó program létrehozása, változtatható mértékegységre váltó program létrehozása, három dobókocka dobás szimuláció, 10000 dobás szimulálása, és vizualizása, különböző jelek előállítás, fázishasítás szimuláció.

Keywords-National Instruments LabVIEW; Virtual Instrument; Front panel; Block panel; subVI;

MÉRÉSSSEL KAPCSOLATOS FOGALMAK

- **LabVIEW:** A National Instruments (NI) által készített Laboratory Virtual Instrument engineering Workbench azaz LabVIEW. Első verziója 1986-ban jelent meg, használata világszerte elterjedt, a világ egyik vezető szoftvere a mérés- és automatizálás területén. A LabVIEW környezetben egy úgynevezett **virtuális műszer (Virtual Instrument, VI)** megvalósítására van lehetőség grafikus programozási környezetben, de általános célú programok fejlesztésére is felhasználható. A grafikus programozási környezet annyit jelent, hogy a programozás során nem szöveges kód készül, hanem különböző függvényeket és vagy utasításokat reprezentáló elemek összekapcsolásával épül fel a program. LabVIEW környezetben folyamat vezérelt, adatfolyam elvű programozásra van lehetőség, a program végrehajtási sorrendjét az utasítások kapcsolódási rendszere határozza meg. A LabVIEW rendelkezik saját fájlformátummal, ez a .vi. A program két fontos része a **Block Diagram** és a **Front panel**. [1] [2]
- **Front panel:** Amikor megnyitunk egy új, vagy létező VI-t, akkor megjelenik a grafikus felület, másnéven a **Front panel**. A jobb egérgomb meg nyomásával jelenik meg a Controls panel. Ebből a menüsorból lehet kiválasztani **kontrolokat**, például egy kapcsolót. A **Front panel** csak kijelzésre szolgál, programozási rész és kapcsolók beállítása nem ezen a felületen történik, arra a **Block Diagram** grafikus felülete szolgál. A program futtatását is ezen a kijelzőn kell elvégezni, a program **kontrolljait** és **indikátorait** is itt lehet kezelni. [3]
A Front Panel grafikus megjelenése az 1. és a 3. ábrákon látható.
- **Block diagram:** Itt történik a **VI** programozása. Itt van lehetőségünk beállítani, hogy a **Front panelel** elhelyezett komponensek, hogy viselkedjenek például egy gomb megnyomása esetén, milyen karakterisztikával rendelkezzenek. A komponenseket vezetékekkel összekötve lehet a program logikai működését előállítani. A jobb egérgomb



1. ábra. A 2. feladat Front Panelje.

megnyomásával megjeleníthető a Functions eszköztár. Itt lehet különböző beépített függvényeket használni. Ilyenek például az operátorok, logikai műveletek, mátrixműveletek, struktúrák, mint a case struktúra, while és for loop és hasonlóak. [4]

A Block Diagram grafikus megjelenése a 2. ábrán látható.

- **Connector panel:** Ha szeretnénk használni egy **VI**-t **subVI**-ként akkor a connector panelen a következő módosításokat kell elvégezni. Minden **VI**-nek van egy ilyen menüje a jobb felső sarokban. Ez arra szolgál, hogy összekössük egy VI kontrolljait, indikátorait egy másik **VI**-vel mint a **subVI** komponens lábai. A konnektor panel, helyes beállításával képesek vagyunk például számokat kinyerni a **subVI**-ból és eltárolni azokat. [5]
- **Indikátor:** Az indikátorok arra szolgálnak, hogy megjelenítsük az adatokat, grafikonokat és egyéb információkat. Ezt a menüpont a **Front panel** jobb egér kattintásával lehet elérni a Controls Palette nevű menüsorban. Amennyiben elhelyezünk egy indikátort a **Front Panelel**, akkor ez a **Block diagramon** is meg fog jellelni. Amennyiben duplán kattintunk az indikátorra a **Front panelel** akkor, automatikusan megnyílik a **Block diagram**, és fordítva. [6]
- **Kontrol:** Olyan grafikus elemek, komponensek, amik a **Front Panelel** és a **Block Diagramon** jelennek meg, amikkel a felhasználó interaktálni tud, és ezzel együtt

az elkészített program paramétereit módosítja. Egy **VI subVI**-ként való használata esetén ezeket bemenetként lehet használni egyenesen a **Block Diagram**ról. [6]

- VI: A **LabVIEW** programban egy függvényként leírható programozható 'virtual instrument' rövidítése.
- SubVI: A **LabVIEW** lehetővé teszi, hogy már elkészített **VI**-ket használjunk más **VI**-kben. Ezt az úgynevezett subVI-k segítségével tudjuk elérni. [7]
- Hisztogram: A hisztogram egy diagram, amely oszlopok sorozatából áll, amelyek magassága vagy hossza az adott kategóriák vagy értéktartományok gyakoriságát vagy eloszlását mutatja. Általában az x-tengelyen az érték tartománya vagy kategóriák vannak megjelenítve, míg az y-tengelyen az értékek vagy kategóriák gyakorisága vagy eloszlása jelenik meg. A hisztogram gyakran használt az adatok vizuális elemzésére és ábrázolására, különösen az adatok eloszlásának megértésére.
- Effektív érték: 'A villamos áram effektív értéke (vagy négyzetes középértéke) az áram hőhatására ad útmutatást. Az effektív érték annak az egyenáramnak az értékével egyenlő, amely azonos idő alatt ugyanakkora munkát végez (hőt termel), mint a vizsgált váltakozóáram.' - [8] Az effektív érték kiszámításának módját a 1. képleten lehet látni.

$$U_{\text{effektív}} = \sqrt{\frac{1}{T} \int_0^T u^2 dt} \quad (1)$$

I. LABVIEW FELADAT

Ebben a feladatban két különálló programot kell létrehozni egy **VI**-ben. Az első egy olyan, amiben egy kontrollt kell összekötni egy indikátorral, amennyiben egy tolokapsolót bekapcsolunk, egy piros ovális LEDnek be kell kapcsolnia. Ennek a kapcsolása a **Block Diagram**mon a következőképpen néz ki. Összekötjük a **kontroll** kapcsolót az **indikátor** LEDdel. A LED ovális mivoltát a **Front panelen** adjuk meg neki, a komponens egyik szélén lévő keretre rákattintva elhúzzuk kifelé.

A második feladat egy mértékegységváltó program megvalósítása, ami m/s-ról váltja km/h-ra a megadott mennyiséget. Ezt a funkciókon belül a 'numeric' menüponton keresztül a 'multiply' komponenssel valósítottuk meg a következő képletbe (2. képlet) helyettesítve.

$$\text{km/h} = \text{m/s} \cdot 3,6 \quad (2)$$

II. LABVIEW FELADAT

A második feladatot az első feladatra építjük. Az eltérés a feladatban az, hogy ebben a feladatban egy háromállású kapcsolóval lehet állítani azt, hogy m/s-ről milyen mértékegységre váltsuk a megadott mennyiséget. Ehhez szükségünk van a 2. és a 3. képletekre.

$$\text{mph} = \text{m/s} \cdot 2,24. \quad (3)$$

Ahhoz, hogy a **Front Panelen** lehessen állítani, hogy milyen mértékegységre váltunk szükségünk van egy több állásra képes **indikátorra**. Ehhez a jobb kattintás által megjelenő 'controls' lebegő menüből a 'numeric' menüpontból a dial komponenset helyezzük el a **Front Panelen**. Ezen a komponensen a következő módosításokat kell elvégezni a megfelelő működés érdekében. A jobb kattintás után a 'properties' gombra kattintva a 'data type'-ot átállítottuk a 'Byte' opcióra, ezzel megszabadulunk a nem egész opciókig. Ezt a típust csak azért

választhattuk, mivel 3 opcióra van szükségünk, és ez belefér a típus által megkülönböztethető értékek számába. A 'scale' menüpont alatt átállítjuk az alapértelmezett minimum és maximum értékeket 1-re és 3-ra. A 'data entry' menüpont alatt kilépünk a 'Use Default Limits' opcióból, majd az 'increment' értékét átállítjuk '1.0'-re, majd a mellette lévő legördülő menüt átállítjuk a 'Coerce to nearest' értékre. Ezzel ez a komponens csak egész értékeket tud elfogadni bemenetnek 1 és 3 között. Miután a **Front Panelt** kellően beállítottuk, a **Block Diagramon** fejeztük be a program funkcionalitását. A dial komponens kimenetét egy case structure-re kötve különböző kódokat tudunk lefuttatni aszerint, hogy milyen érték van a case structur-re kötve. Ez a mi esetünkben a dial kimenete. Ebbe a case structure-be a mértékegység váltásához szükséges konstansokat rakjuk bele, amit a case structure-ön kívül szorzunk meg a megadott m/s mennyiséggel.

Utolsó lépésként a mértékegység váltásért felelős részét a **VI**-nek egy **subVI**-be építjük át. Itt fontos lépés a jobb felső sarokban a **kontroll** és **indikátorok** megosztása a **subVI** komponens lábaihoz.

III. LABVIEW FELADAT

Ebben a feladatban egy olyan programot hozunk létre, ami egy gomb nyomásra három 6 oldalú dobókocka dobást szimulál, majd ezeknek az eredményét kiírjuk három szám **indikátorra**.

Miután minden szükséges indikátort és kontrollert elhelyeztünk a **Front Panelen** (ezek egy gomb, három szám indikátor és egy LED), majd a címkéket átirítottuk, hogy értelmezhető legyen a front panel, és a gomb karakterisztikáját átállítjuk úgy, hogy lenyomva csak egyszer adjon ki logikai igaz értéket. A program felépítését a **Block Diagram** folytatjuk. A gomb logikai értékét egy switch structure-re kötve igaz érték esetén random számot adunk vissza az $[1; 6] \in \mathbb{N}$ intervallumon, ez reprezentálja egy dobókocka véletlenszerű dobását. Ezt az értéket az adott szám **indikátorra** kiírjuk. A kódnak ezt a részét háromszor helyezzük el a **Block Diagram**mon, a három egyidejű dobás miatt. Ezeket a számokat összeadjuk a case structure-ön kívül, és megnézzük, hogy ez egyenlő-e 18-al. Amennyiben igen a **Front Panelen** elhelyezett LEDet bekapcsoljuk. Itt figyelni kell arra, hogy a case structure akkor is adja meg a három értéket, amikor a gomb értéke hamis. Itt erre a célra a 0-át választottuk, hiszen dobókockával nem lehet ezt a számot dobni.

Az egyes esélyekre, hogy a három összeg szummájának értéke mekkora, fontos tudnunk, hogy az adott számot hányféleképpen tudjuk összeadni 3 1-től 6-ig terjedő szám segítségével. Ezek a következők:

$$3 = 1 + 1 + 1 \quad (1 \text{ féle képpen})$$

$$4 = 1 + 1 + 2 \quad (3 \text{ féle képpen})$$

$$5 = 1 + 1 + 3 = 1 + 2 + 2 \quad (6 \text{ féle képpen})$$

$$6 = 1 + 1 + 4 = 2 + 2 + 2 = 1 + 2 + 3 \quad (10 \text{ féle képpen})$$

$$7 = 1 + 1 + 5 = 1 + 2 + 4 = 1 + 3 + 3 = 2 + 2 + 3 \quad (15 \text{ féle képpen})$$

$$8 = 1 + 1 + 6 = 1 + 2 + 5$$

$$8 = 1 + 3 + 4 = 2 + 2 + 4 = 2 + 3 + 3 \quad (21 \text{ féle képpen})$$

$$9 = 1 + 2 + 6 = 1 + 3 + 5 = 1 + 4 + 4 = 2 + 2 + 5$$

$$9 = 2 + 3 + 4 = 3 + 3 + 3 \quad (25 \text{ féle képpen})$$

$$10 = 1 + 3 + 6 = 1 + 4 + 5 = 2 + 2 + 6 = 2 + 3 + 5$$

$$10 = 2 + 4 + 4 = 3 + 3 + 4 \quad (27 \text{ féle képpen})$$

$$11 = 1 + 4 + 6 = 1 + 5 + 5 = 2 + 4 + 5 = 3 + 3 + 5$$

$$11 = 4 + 4 + 3 = 2 + 3 + 6 = \quad (27 \text{ féle képpen})$$

$$12 = 6 + 5 + 1 = 4 + 3 + 5 = 4 + 4 + 4 = 5 + 2 + 5$$

$$12 = 6 + 4 + 2 = 6 + 3 + 3 \quad (25 \text{ féle képpen})$$

$$13 = 6 + 6 + 1 = 5 + 4 + 4 = 3 + 4 + 6 = 6 + 5 + 2$$

$$13 = 5 + 5 + 3 \quad (21 \text{ féle képpen})$$

$$14 = 6 + 6 + 2 = 5 + 5 + 4$$

$$14 = 4 + 4 + 6 = 6 + 5 + 3 \quad (15 \text{ féle képpen})$$

$$15 = 6 + 6 + 3 = 6 + 5 + 4 = 5 + 5 + 5 \quad (10 \text{ féle képpen})$$

$$16 = 6 + 6 + 4 = 5 + 5 + 6 \quad (6 \text{ féle képpen})$$

$$17 = 5 + 6 + 6 \quad (3 \text{ féle képpen})$$

$$18 = 6 + 6 + 6 \quad (1 \text{ féle képpen})$$

Ezeknek a tudatában behelyettesíthetünk az általános valószínűség-számítás képletbe. Ez a következőféleképpen néz ki (4. képlet):

$$p = \frac{k}{\ddot{o}}, \quad (4)$$

ahol k a kedvező lehetőségek számát, míg \ddot{o} az összes lehetőségek számát jelöli. Az összes lehetőség száma a következőképp számolható ki (5. képlet):

$$\ddot{o} = 6^3 = 216. \quad (5)$$

Az előző sorokban kiszámolt kedvező lehetőségek alapján kiszámolható az adott számok valószínűsége három hatoldalú dobókocka dobás után kapott számok összegeként:

$$p(3) = \frac{1}{216} \approx 0,004\%,$$

$$p(4) = \frac{3}{216} = \frac{1}{72} \approx 0,01389\%,$$

$$p(5) = \frac{6}{216} = \frac{1}{36} \approx 0,0278\%,$$

$$p(6) = \frac{10}{216} = \frac{5}{108} \approx 0,0463\%,$$

$$p(7) = \frac{15}{216} = \frac{5}{72} \approx 0,069\%,$$

$$p(8) = \frac{21}{216} = \frac{7}{72} \approx 0,097\%,$$

$$p(9) = \frac{25}{216} \approx 0,115\%,$$

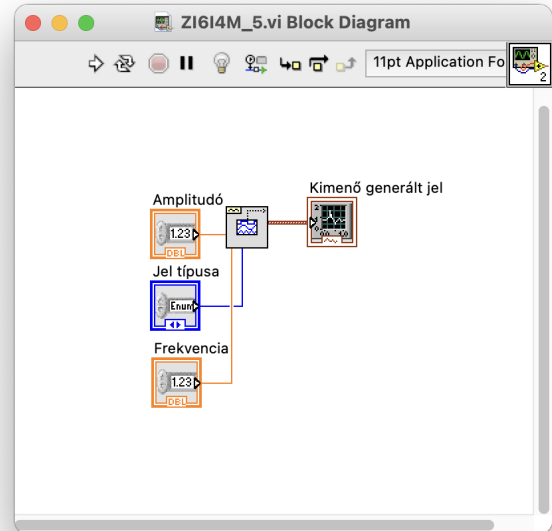
$$p(10) = \frac{27}{216} = \frac{1}{8} \approx 0,125\%,$$

$$p(11) = \frac{27}{216} = \frac{1}{8} \approx 0,125\%,$$

$$p(12) = \frac{25}{216} \approx 0,115\%,$$

$$p(13) = \frac{21}{216} = \frac{7}{72} \approx 0,097\%,$$

$$p(14) = \frac{15}{216} = \frac{5}{72} \approx 0,069\%,$$



2. ábra. A félkész 5. feladat Block Diagramja.

$$p(15) = \frac{10}{216} = \frac{5}{108} \approx 0,0463\%,$$

$$p(16) = \frac{6}{216} = \frac{1}{36} \approx 0,0278\%,$$

$$p(17) = \frac{3}{216} = \frac{1}{72} \approx 0,01389\%,$$

$$p(18) = \frac{1}{216} \approx 0,004\%.$$

IV. LABVIEW FELADAT

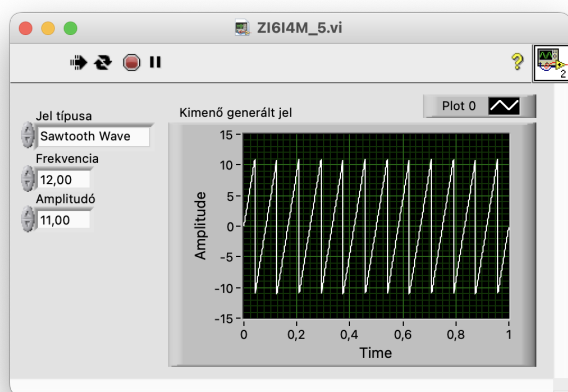
A negyedik feladat a III. fejezetben elkészített VI-t kell subVI-ként használni. Ehhez a már meglévő III. feladatot kell átalakítani, a **kontrol** és **indikátor** komponenseket bemeneteknek és kimeneteknek beállítani. Ezt követően ezt egy 10000 iterációs for loopba helyezni. Az eredmények megjelenítésére használhatunk **hisztogrammot**. Ennél be kell állítani a várható minimum és maximum értékeket, illetve azt is, hogy hány számot várhatunk, hány oszlopot mutasson a megjelenő **hisztogram**.

V. LABVIEW FELADAT

Ebben a feladatban egy olyan VI-t kell megalkotnunk, amely képes különböző jelek előállítására. Ezek a szinusz, négyszög, fűrész és a háromszög jelek. A programnak ezen kívül rendelkeznie kell három **kontrollal**, amiken állítvány az előállított jel amplitudója, offszete és frekvenciája. Ezt a programot nem tudjuk befejezni, mivel kifutottunk a labor időkeretéből. A program aktuális **Block Diagramja** és **Front Panelje** a 2. és a 3. ábrákon láthatóak.

VI. LABVIEW FELADAT

A hatodik feladatban egy hasítási pont szimulációt kellett elkészíteni, amivel a bekapcsolási pillanat változásával nézzük meg a kimeneti teljesítményt, és annak az **effektív értékét**. A feladatot nem tudtuk a gyakorlaton befejezni, mivel sajnos a mérőpárral kifutottunk az időből.



3. ábra. A félkész 5. feladat Front Panelje.

HIVATKOZÁSOK

- [1] “Bevezetés a LabVIEW világába”. (), cím: https://megtestesules.info/hobbielektronika/2019/labview19_01.pdf.
- [2] F. Gergely. “Friedl Gergely- LabVIEW Segédlet”. (), cím: https://maxwell.sze.hu/~friedl/Szab%C3%A1lyoz%C3%A1si_rendszerek/LabVIEW%20seg%C3%A9dlet.pdf.
- [3] N. Instruments. “LabVIEW Supplemental - LabVIEW Front Panel Explained”. (), cím: <https://www.ni.com/en/support/documentation/supplemental/08/labview-front-panel-explained.html>.
- [4] N. Instruments. “LabVIEW Supplemental - LabVIEW Block Diagram Explained”. (), cím: <https://www.ni.com/en/support/documentation/supplemental/08/labview-block-diagram-explained.html>.
- [5] N. Instruments. “LabVIEW Fundamentals - Building the Connector Pane”. (), cím: <https://www.ni.com/docs/en-US/bundle/labview/page/building-the-connector-pane.html>.
- [6] L. M. Hub. “LabVIEW Basics - 03 | Controls and Indicators”. (), cím: https://www.labviewmakerhub.com/doku.php?id=learn:tutorials:labview:basics:controls_indicators.
- [7] Digilent.com. “Creating a subVI in National Instruments LabVIEW”. (), cím: <https://diligent.com/blog/creating-a-subvi-in-labview/>.
- [8] “Effektív Érték.pdf (PPKE ITK Bevezetés a mérés technikába és jelfeldolgozásba)”.