Chapter 5

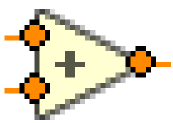
MEASUREMENT AND ANALYSIS WITH LABVIEW SOFTWARE

Ahmet ÖZMEN[1], *Aykut COŞKUN*[2],
Mehmet ERTUĞRUL[3]

1 Department of Electronics and Automation, Vocational School, Ağrı İbrahim Çeçen University, Ağrı, Turkey, 0000-0002-3631-4883, ahmet25440@gmail.com
2 Department of Electronics and Automation, Vocational School of Technical Sciences, Bayburt University, Bayburt, Turkey, 0000-0002-7240-6865
3 Department of Electrical and Electronics, Engineering Faculty, Atatürk University, Erzurum, Turkey, 0000-0003-1921-7704

INTRODUCTION TO LABVIEW PROGRAM

The LabVIEW program consists of two screens. These screens are Block diagram and Front panels. The block diagram section is the part where the programmer works. That is, this is the part where the programmer creates the operations to be performed depending on the algorithm. The front panel is the screen that will appear to the user when the algorithm runs. Block diagram can be considered as coding part. Programming by combining blocks without writing code is the convenience of LabVIEW program. This section contains the content that the programmer needs such as mathematical operations, control structures, textual operations, file reading, writing and recording, network operations, database operations, libraries of loaded devices. The front panel is the part of the program to be shown to the user.



In the **Block Diagram**, there are icons corresponding to the operation we want to do. These icons are encoded and made available to us by LabVIEW programmers. These icons are basically composed of two parts: input and output. Using these parts, the desired operation is performed depending on our algorithm. For example, let's create a program, which shows the sum with a fixed value of a number entered externally. To do this, right-click on the block diagram and select "**Add"** from the "**Numeric**" section as shown in Figure 1. As shown in the collection process icon, there are connection points around the icon. These points will be used to determine our flowchart that will form our algorithm. Right-click on the part you want to connect to and click "**Create**". Here, 3 parts will be observed as shown in Figure 2. These are "**Constant, Control, Indicator**" sections. Selecting

Constant allows us to enter a constant value in the Blog diagram. When **Control** is selected, it creates an icon on the Front panel and allows the user to intervene in this section. These two operations can be applied to the input part of the icon. **Indicator** is applied to the output part of the icon. This will display the result of the action on the icon in the Front panel. These three sections are the most commonly used parts of the LabVIEW program. In addition, the output of one icon may be the input of another icon. To do this, left-click one of the parts to be connected. Operation is performed by moving the mouse and right-clicking on the port of the other icon again.

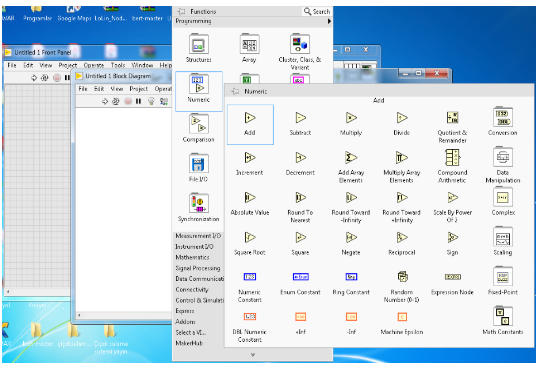


Figure 1. *LabVIEW Program Collection Process*

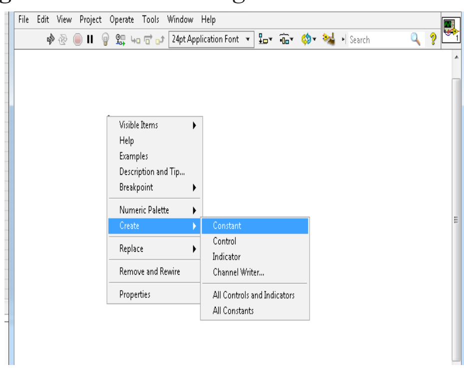


Figure 2. *Create Process*

CONTROL THE KEITLEY DEVICE WITH THE LABVIEW PROGRAM

Many devices communicate easily with LabVIEW (Academy, Sciences, & Academy, 2013; Akgül, 2017; Ballesteros, Fernández Palop, Hernández, Crespo, & del Pino, 2004; Bohórquez, Enrique Gómez, & Andújar Márquez, 2009; Boutana, Mellit, Haddad, Rabhi, & Pavan, 2017; Chouder, Silvestre, Taghezouit, & Karatepe, 2013; Demirbaş & Bayhan, 2009; Forero, Hernández, & Gordillo, 2006; Hamel & Mohellebi, 2020; Hu et al., 2020; Kobayashi, Takano, & Sawada, 2006; ŞAHİN, BAYRAKTAR, KAVUR, & EVREN ŞAHİN, 2017; Sziki, Sarvajcz, Szántó, & Mankovits, 2019; Technology, 2010; Zhang et al., 2020). Especially the KEITLEY device used in laboratories will be controlled with LabVIEW. Other devices are controlled in a similar way to this device. In this process, we need to install LabVIEW drivers of our device. This can be done in several different ways. First, the device name is added from the NI LabVIEW website by typing and searching. Similarly, the program is loaded from the device manufacturer's website or on the CD supplied with the device. In another method, "**Help / Find Instrument Drivers...**" is selected from the LabVIEW software.

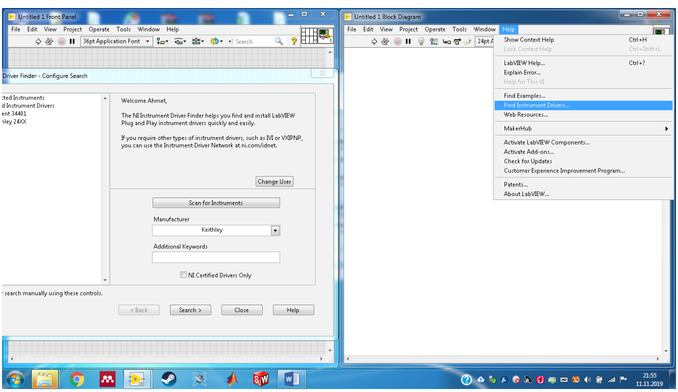


Figure 3. *Adding Drivers*

The "**NI Instrument Driver Finder - Configure Search**" window will open as shown in Figure 3. In this section, if you want a user name and password, you can register for free on the **NI Instrument** home page. **Keithley** is selected from **Manufacturer** in this window. In this section, the name of the device we want to use or the name of the company is selected. The model of the device is selected from Additional Keywords. Then click on "Search" to find the drivers from the database. Found drivers are installed by selecting and clicking "**Install**".

Installed device VIs are located in **Instrument Drivers** section of **Instrument I / O** section by right clicking on **Block Diagram** as shown in Figure 4. In this section, we have VIs for all the devices we have installed.

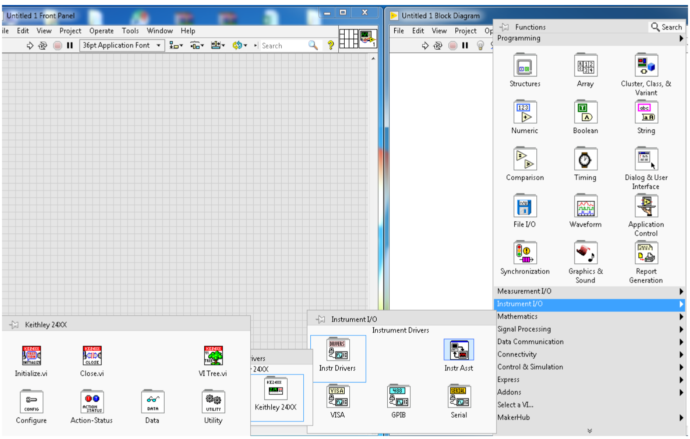


Figure 4. *Installed Drivers*

Now that all this is complete, we can now create our application. To control a device, basically our computer and software need to speak the same language as this device. Now that we've installed this plug-ins, let's look at the first VI, **Initialize.vi**. As shown in Figure 5, this VI will allow us to select the way in which the devices communicate and to make the settings of this communication protocol. These communication paths are a protocol such as **VISA, BPIB and Serial** and are determined according to the device we have. The device is connected to our computer with the connection cable according to this protocol. After seeing the device on the computer, it will appear in the **VISA** resource name of Initialize.vi. First of all, we should right-click the VISA resource name and select **Create / Control** so that we can see the link in the Front Panel. After seeing the port to which our device is connected, we need to set the other connection settings to match the connection settings of the device. If these settings are not known, we will need to learn from the device user manual or the connection information section of the device's own menus.

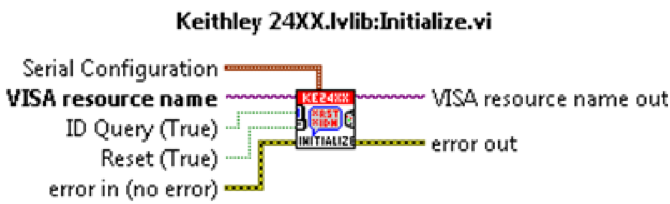


Figure 5. *Initialize.vi*

Let us not forget that in this way we can communicate with all devices with LabVIEW drivers. Now, we can make measurements, we can change the device parameters, we can check the input and output ports of the device, we can perform mathematical operations, we can analyze, we can draw graphs, we can save our data to our computer. The LabVIEW program allows a large number of analyzes similar to these procedures. As an example, let's measure voltage from our Keithley device. To do this, open the screen in Figure 4 and select **Read.vi** from the **Data** section. As shown in Figure 6, the outputs at the top right and bottom of **Initialize.vi** are connected to the inputs at the top left and bottom of Read.vi, and the outputs are terminated by connecting to **Close.vi**. This process is basically done in this way in such applications. Generally, the connections at the bottom are used to identify and show the errors that occur in the system. The connections on the upper side are generally used to transfer the information to the next VI. In addition, **While Loop** is used as shown in Figure 6. This is to repeat the measurement for one cycle. **While Loop** is selected as shown in Figure 7, the part of the loop to be created by holding down the left mouse button is selected and released. A delay is added for the cycle repeat time, and finally a button is added the red dot on the bottom left cause the loop to end by clicking.

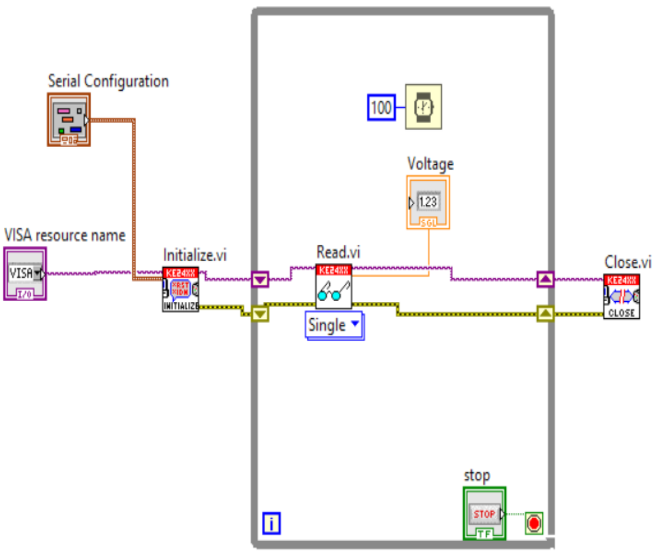


Figure 6. *Voltage Reading*

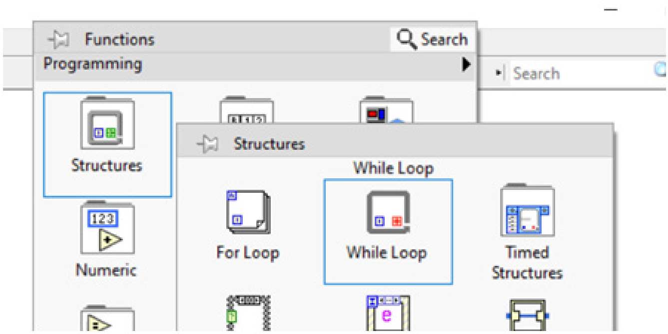


Figure 7. *While Loop*

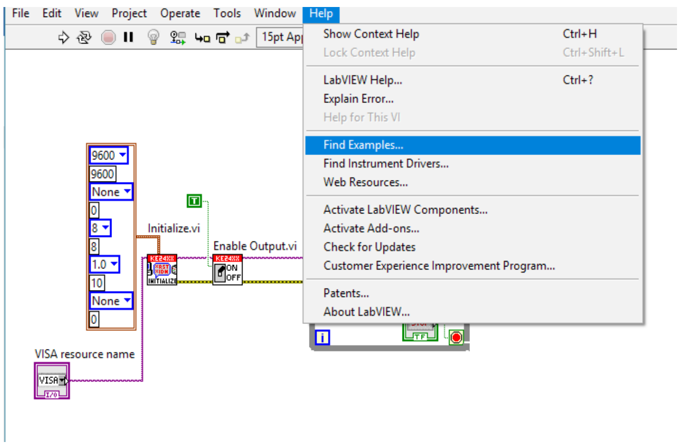


Figure 8. *Example applications*

The sample applications of many devices that we will use are available in the LabVIEW program. We can use these sample applications or develop our own application by examining them. Sample applications can be accessed the “Find Examples…” section of the “Help Menu” in Figure 8.

RESULT

As can be seen, it can be controlled from a device in a very short time by using LabVIEW software without the need of any code information. In this way, both we can easily create our own systems and we can save by analyzing our data in computer environment. In addition, a complex system of several devices and sensors can run synchronously with each other in a single software environment. In this way, if the operation of one device depends on the result produced by another device, we can do it in a very short time in a single software environment.

REFERENCES

Academy, T. N., Sciences, N., & Academy, T. N. (2013). Labview Based Target Recognition And Tracking System. *Deniz Bilimleri ve Mühendisliği Dergisi*, *9*(2), 66–71.

Akgül, A. (2017). Elektronik devrelerin gerçek zamanlı bode diyagramlarının LabVIEW ile elde edilmesi. *SAÜ Fen Bilimleri Enstitüsü Dergisi*, 1–1. https://doi.org/10.16984/saufenbilder.310655

Ballesteros, J., Fernández Palop, J. I., Hernández, M. A., Crespo, R. M., & del Pino, S. B. (2004). LabView virtual instrument for automatic plasma diagnostic. *Review of Scientific Instruments*, *75*(1), 90–93. https://doi.org/10.1063/1.1634356

Bohórquez, M. A. M., Enrique Gómez, J. M., & Andújar Márquez, J. M. (2009). A new and inexpensive temperature-measuring system: Application to photovoltaic solar facilities. *Solar Energy*, *83*(6), 883–890. https://doi.org/10.1016/J.SOLENER.2008.12.007

Boutana, N., Mellit, A., Haddad, S., Rabhi, A., & Pavan, A. M. (2017). An explicit I-V model for photovoltaic module technologies. *Energy Conversion and Management*, *138*, 400–412. https://doi.org/10.1016/J.ENCONMAN.2017.02.016

Chouder, A., Silvestre, S., Taghezouit, B., & Karatepe, E. (2013). Monitoring, modelling and simulation of PV systems using LabVIEW. *Solar Energy*, *91*, 337–349. https://doi.org/10.1016/J.SOLENER.2012.09.016

Demirbaş, Ş., & Bayhan, S. (2009). Güç sistemlerinde harmoniklerin gerçek zamanli ölçüm ve analizi. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *24*(3), 461–468.

Forero, N., Hernández, J., & Gordillo, G. (2006). Development of a monitoring system for a PV solar plant. *Energy*

Conversion and Management, *47*(15–16), 2329–2336. https://doi.org/10.1016/J.ENCONMAN.2005.11.012

Hamel, M., & Mohellebi, H. (2020). A LabVIEW-based real-time acquisition system for crack detection in conductive materials. *Mathematics and Computers in Simulation*, *167*, 381–388. https://doi.org/10.1016/J.MATCOM.2018.02.004

Hu, Y., Wang, T., Chen, T., Song, N., Yao, K., & Luo, Y. (2020). Design and implementation of testing system of LED driver power based on LabVIEW. *Optik*, *200*, 163411. https://doi.org/10.1016/J.IJLEO.2019.163411

Kobayashi, K., Takano, I., & Sawada, Y. (2006). A study of a two stage maximum power point tracking control of a photovoltaic system under partially shaded insolation conditions. *Solar Energy Materials and Solar Cells*, *90*(18–19), 2975–2988. https://doi.org/10.1016/J.SOLMAT.2006.06.050

ŞAHİN, S., BAYRAKTAR, M., KAVUR, A. E., & EVREN ŞAHİN, K. (2017). Arduino ve LabVIEW Kullanarak EMG Verilerinden Eşik Seviye Belirleme ile Motor Kontrol Düzeneği Tasarımı. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *22*(2), 736. https://doi.org/10.19113/sdufbed.06905

Sziki, G. Á., Sarvajcz, K., Szántó, A., & Mankovits, T. (2019). Series Wound DC Motor Simulation Applying MATLAB SIMULINK and LabVIEW Control Design and Simulation Module. *Periodica Polytechnica Transportation Engineering*, *48*(1), 65–69. https://doi.org/10.3311/pptr.12908

Technology, C. (2010). *ELEKTRONİK DENEY MODÜLLERİNİN LabVIEW İLE KONTROLÜ*. *2*(3), 1–8.

Zhang, Y., Li, Y., Gu, X., Liu, H., Zhang, Y., & Hu, W. (2020). Research on measurement of laser beam M2 factor based on LabVIEW. *Optik*, *203*, 163759. https://doi.org/10.1016/J.IJLEO.2019.163759