iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** II **Month of publication:** February 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Implementation of Smart Laboratory Using Lab VIEW

M. Sridevi[1], Student Member, IEEE
[1]PG Student, Department of ECE,
Sri Krishna College of Engineering and Technology, Coimbatore

Abstract : Smart laboratory implementation is a developing complex technology. It needs a powerful data acquisition system that continuously monitors parameters that involving security and environment safety in laboratory and to acquire, process, analyze and communicate parameters to a centralized data center using NI LabVIEW software and NI-DAQmx driver software. This paper presents the hardware implementation of a multiplatform control system for laboratory automation using LabVIEW. A sample laboratory environment monitoring and control system that is one branch of the automation addressed in this paper. The database of parameters such as temperature, humidity voltage, current, alarms, smoke sensors and other environmental conditions in the laboratory is created and monitored. This system can be connected to internet to monitor and control the laboratory equipment's from anywhere in the world. The approach combines hardware and software technologies.
Index Terms: NI-DAQmx, voltage sensor proximity sensor, Burglar alarm/Smoke detector

I. INTRODUCTION

A. LabVIEW

LabVIEW – an acronym for Laboratory Virtual Instrumentation Engineering Workbench is a system design platform and development environment for a visual programming language from National Instruments.

B. Dataflow programming

The programming language used in LabVIEW, also referred to as G programming (Graphical programming), is a dataflow programming language. Execution is determined by the structure of a graphical block diagram (the LV-source code) on which the programmer connects different function-nodes by drawing wires. These wires propagate variables and any node can execute as soon as all its input data become available. Since this might be the case for multiple nodes simultaneously, G programming is inherently capable of parallel execution. Multi-processing and multi-threading hardware is automatically exploited by the scheduler, which multiplexes multiple OS threads over the nodes ready for executions. LabVIEW ties the creation of user interfaces (called front panels) into the development cycle. LabVIEW programs/subroutines are called virtual instruments (VIs). Each VI has three components:

1) Block diagram

2) Front panel

3) Connector panel

The connector panel is used to represent the VI in the block diagrams of other, calling VIs. Controls and indicators on the front panel allow an operator to input data into or extract data from a running virtual instrument. However, the front panel can also serve as a programmatic interface. Thus a virtual instrument can be run in two ways either be run as a program, with the front panel serving as a user interface, or, when dropped as a node onto the block diagram. The front panel defines the inputs and outputs for the given node through the connector panel. This implies each VI can be easily tested before being embedded as a subroutine into a larger program.

C. OS Support

LabVIEW source code and development is supported by Windows 9x/2000/NT/XP, Apple Macintosh (including X), Power ax OS, Solaris, HP-Unix, Sun, Linux, the Pharlap RTOS, and VxWorks RTOS (Real-Time Operating Systems, found on National Instruments embedded controllers). Code developed under one platform can be ported to any of the others, recompiled and run. LabVIEW can run on handheld devices, such as Microsoft Windows Mobile for Pocket PC devices.

D. Interfacing

The key benefit of LabVIEW over other development environments is the extensive support for accessing instrumentation hardware. It offers standard software interfaces to communicate with hardware devices. The provided driver interfaces save program development time. A new hardware driver topology (DAQmx Base) provides platform independent hardware access to numerous data acquisition and instrumentation devices. The DAQmx Base driver is available for LabVIEW on Windows, Mac OS X and Linux platforms. We can use LabVIEW to communicate with hardware such as data acquisition, vision, and motion control devices, and GPIB, PXI, VXI, RS-232, and RS-484 devices. Lab.

II. HARDWARE DESCRIPTIONS

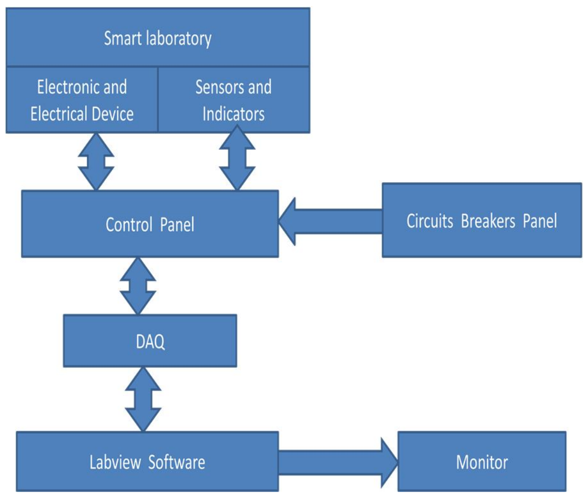


Fig.1: Block Diagram

The analog signals measured from the sensor circuits are acquired using DAQmx interfaced with LabVIEW software, and converted into measurable parameters, thus environmental and security conditions in the laboratory are monitored.

III. BRIEF METHODOLOGY

The project is designed with NI Daqmx, Voltage sensors, Temperature sensor, Proximity sensor and Burglar alarm/Smoke detector

A. NI-DAQmx Features

National Instruments provides valuable measurement services software with data acquisition devices. This measurement services software reduces the time costs that make up two-thirds of the total costs associated with data acquisition and logging application creation.

B. NI-DAQ mx Simulated Device

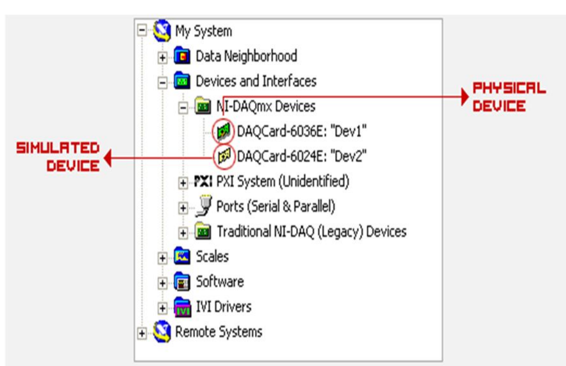


Fig.2 :NI-DAQmx Simulated Device

NI-DAQmx simulated devices are useful for creating and running NI- DAQmx programs and for trying out tools such as the DAQ

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Assistant or NI LabVIEW Signal Express without using any physical hardware. We can also use NI-DAQ mx simulated devices for discovering device capabilities without the physical hardware because we can verify NI-DAQmx tasks on simulated devices just as we would on real devices. If a property is set to an invalid value, the error returned for a simulated device is identical to the error returned for a real device. Like real devices, NI-DAQmx simulated devices count and reserve all necessary task resources, such as RTSI lines, PXI trigger lines, DMA channels, counters, and more.

C. DAQ Assistant Express VIO

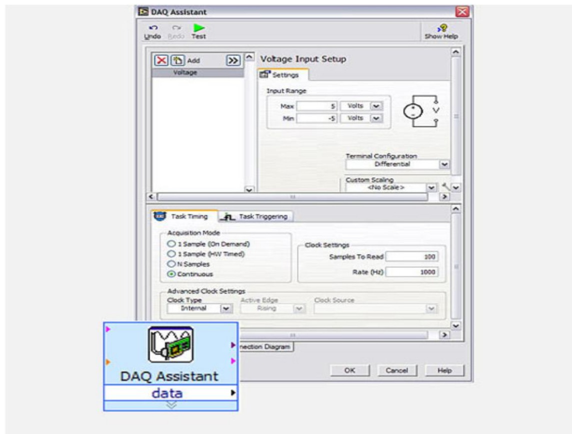


Fig .3: DAQ Assistant Express VIO

The DAQ Assistant, bundled with NI-DAQmx for Windows, provides a step-by-step guide for configuring, testing, and programming measurement tasks. We can also automatically generate example programs based on your configuration for low-level customization. In addition to other configuration-based VIs in LabVIEW, the DAQ Assistant Express VI makes it easier and faster to develop data acquisition applications.

IV. RESULTS

A. Temperature sensor

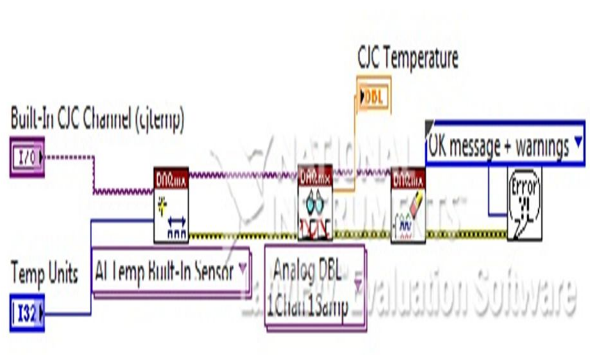


Fig .4: Temperature sensor

The basic element in temperature system is the reading of temperature value from temperature sensor. This is simulation for temperature to be measured inside the laboratory in order to maintain the human comfort range inside the laboratory.
Steps:

1) Create an analog input channel for the built in temperature sensor. Select the units in which the temperature will be returned.
2) Read the temperature value.
3) Call the clear VI task to clear the task.
4) Use the popup dialog box to display the errors.

B. Proximity sensor

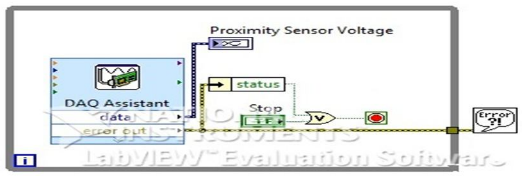


Fig.5: Proximity sensor

The design of motion system used in smart laboratory system involves DAQmx to acquire the motion and sends input signal and thus further data processing and controlling is done by LabVIEW.

Steps:

1) *Get the voltage from proximity sensor through DAQmx.*
2) *Display the voltage.*
3) *Stop if error occurs or user presses stop button.*
4) *Handle the errors if occurred.*

C. Burglar alarm

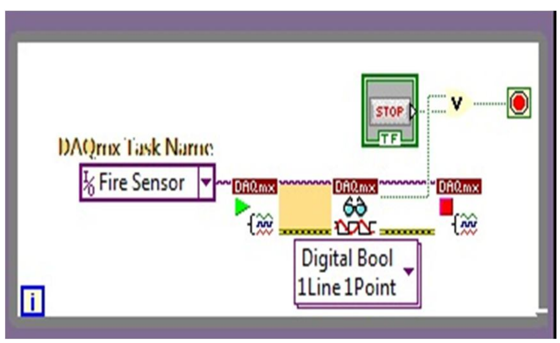


Fig .6: Burglar alarm

The design of Burglar alarm system used in smart laboratory system is similar to the design used for fire alarm system. It is divided into three parts; the first part is the signal that reaches from burglar alarm sensors when its trigger threshold has been reached after any a specific danger in the house. The second part is the output signals that send after the processing of input signal, and final part is the controlling system and data processing by LabVIEW.

D. Voltage Sensor

1) *Ac voltage measurement*

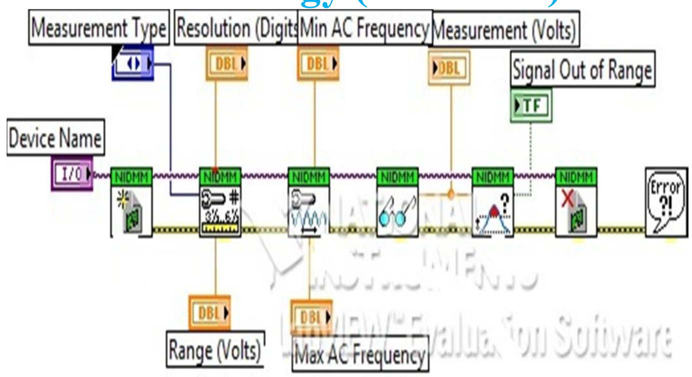


Fig .7:Ac voltage measurement

The ac voltage signal acquired from the voltage sensor is acquired using DAQmx and the analog input signal is converted into measurable form using Lab VEW.

Steps:

a) Connect the input ac voltage device.
b) Configure function, resolution and range.
c) Configure the band width of the AC measurement.
d) Displays an error if any.

2) DC voltage measurement

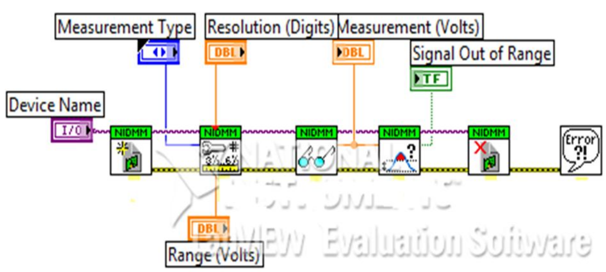


Fig .8:DC voltage measurement

The DC voltage signal acquired from the voltage sensor is acquired using DAQmx and the analog input signal is converted into measurable form using Lab VEW.

Steps:
a) Connect the input DC voltage device.
b) Configure function, resolution and range.
c) Initiate the DMM and return a measured value to the user.
d) Check if the measurement is over ranged.
e) Display the error if any.

V. CONCLUSIONS

Thus by arranging all sensors required for the measurement of every parameter and programming in LabVIEW, smart laboratory system can be implemented. The sensors are interfaced with software using DAQmx in LabVIEW and monitored using PC with LabVIEW.

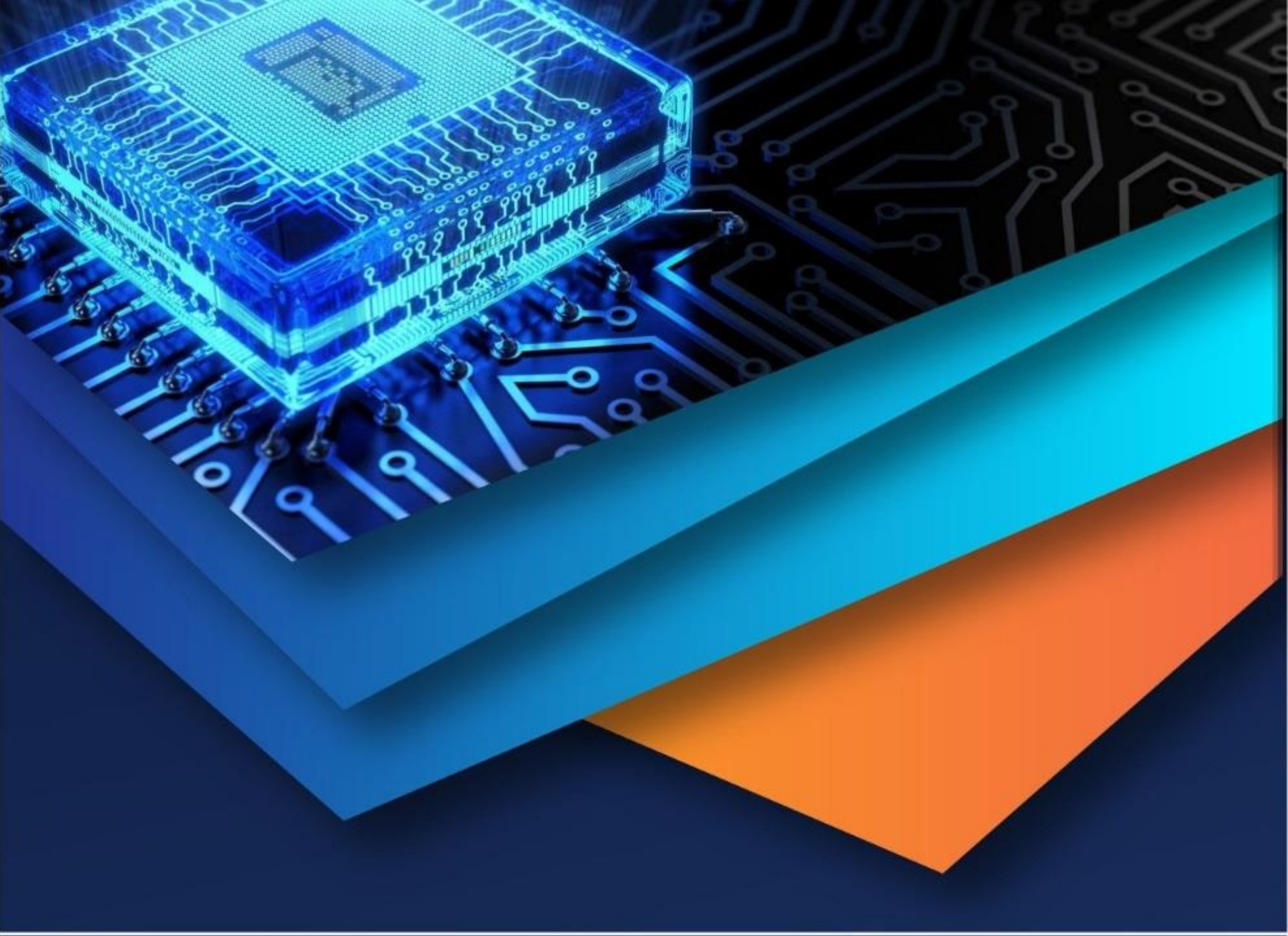
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

VI. FUTURE WORK

This project is further extended by creating a controlled hardware structure from measured input signals from the sensor, creating data log for the measured parameters and connecting the control in sever to operate them in world wide.

REFERENCES

[1] Design and implementation of smart home using Lab VIEW by Basil Hamed
[2]http://depts.washington.edu/dmgftp/publications/html/smarthouse98-mdg.html
[3] Sleman, A.; Alafandi,M.Moeller,"Integration of Wireless Fieldbus and Wired Fieldbus for Health Monitoring"; R.Consumer Electronics, 2009. ICCE '09. Digest of Technical Papers International Conference on 10-14 Jan. 2009
[4] Van Nguyen, T.; Jin Gook Kim; Deokjai Choi, "ISS: The Interactive Smart home Simulator," Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on , vol.03, no., pp.1828- 1833, 15-18 Feb. 2009
[5] Escoffier, C.; Bourcier, J.; Lalanda, P.; Jianqi Yu, "Towards a Home Application Server," Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, vol., no., pp.321-325, 10-12 Jan. 2008.
[6] Salvador, Z.; Jimeno, R.; Lafuente, A.Larrea, M.; Abascal, J.;"Architecture for ubiquitous environmentss" Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on, Volume 4, 22-24 Aug. 2005 Page(s):90 - 97 Vol. 4.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY