Visual Programming of an Interactive Smart Home Application using LabVIEW

Apostolos Meliones and Dimitrios Giannakis
Department of Digital Systems, University of Piraeus, Greece
meliones@unipi.gr, giannakis.dimitrios@gmail.com

***Abstract*—This paper presents the modeling, functionality and implementation of a modern interactive smart home system, which supports several automations for comfortable and secure living, including an energy management subsystem which cooperates with a rooftop photovoltaic installation. The smart home application was developed in the LabVIEW graphical programming environment, being a challenging task since LabVIEW is primarily targeting data collection and analysis as well as complex measurements and trials in laboratory and industrial environments. However, a lot of attractive capabilities it provides for interfacing and control, processing of sensor input, visual (also known as dataflow or diagrammatic) programming and handy visualization promise a much wider adoption and application field. Consequently, it can be applied efficiently and effortlessly in modern home automation and smart home applications.**

***Keywords*— Smart home, home automation, visual programming, LabVIEW, home energy management**

I. Introduction

Recent advances in Intelligent Environments research give a glimpse into the future and reveal exciting visions of smart everything – smart cities, smart homes, smart workplaces, smart hotels, smart schools, and much more. Driven by technological evolution offering low power many-things and wireless almost-everything, we could, in only a decade, envision and prototype impressive cyber-physical systems and applications. The fundamental generalities and necessary ecologies that can lead to better design, development, operation and adaptation of Intelligent Environments are already in place [1]. With the advent of sensor networks and pinhead-size computers, we're moving much closer to realizing the vision of ubiquitous and pervasive computing. However, as we create pervasive spaces, we must think ahead to consider how we program them, just as we successfully programmed the mainframe and, later on, the Internet. In [2], it is stated that programmability of pervasive spaces may involve a graphical development environment in which services, context, and applications are represented or created graphically using a LabVIEW-style building-block interface [3].

Creating the smart home is a reality which will pervade progressively the modern way of living. Technology which is available at home is rapidly advancing, while at the same time user requirements continue rising at a similar or faster pace. There are certain key factors in adopting the smart home in daily routine: security, comfort and assisted living, which is continuously becoming more important for elderly and people with specific needs, and energy saving, e.g. for lighting control based on luminance or preventing needless operation of air conditioning. A smart home can implement a wealth of scenarios involving the centralized or autonomous management of various home subsystems, with the ability to control, either in-house or remotely, various home electrical and mechanical installations, appliances and devices, including multimedia devices, using computers and automation processes.

Several smart home development efforts have followed a visual or diagrammatic programming approach. An early effort to implement smart home control using a LabVIEW server and a wireless PDA client is presented in [4]. Another solution to transform a normal house in a smart house while reducing the energy consumption, realized with the help of wireless sensor networks and of the LabVIEW graphical programming environment is presented in [5]. An energy conservation module which is comprised of a PIC microcontroller and a LabVIEW display which helps in reducing the energy consumption of household appliances, such as lights, is also presented in [6]. Quite similarly, a multiplatform house environment monitor and control system based on LabVIEW is addressed in [7].

This paper presents the modeling, functionality and implementation in the LabVIEW environment of a modern interactive smart home system, which supports several automations for comfortable and secure living, including an energy management subsystem which cooperates with a rooftop photovoltaic installation. It comprises a simplified alternative framework for smart home development as compared to middleware-related efforts, such as e.g. in [8]. Section 2 presents the smart home system requirements and functionality. Section 3 presents the LabVIEW development environment and the virtual instruments which were used for the implementation of the smart home application. Section IV presents the visual development and diagrammatic programming of the smart home application, while section V completes this work with important conclusions.

II. Smart Home System Functionality

The smart home system involves a central controller which is a typical computer that interfaces sensors, actuators and home devices, featuring the following key functionality meeting user requirements:

- Performing of user-configurable automated control operations in supervised spaces taking into account the status of various parameters.

978-1-4799-0752-6/13/$31.00 ©2013 IEEE

- Cancelation of automated control in case of user intervention.
- Displaying of the operational status of all smart home subsystems and devices and any abnormal situation on a central control panel, through a message whiteboard and proper indicator leds. The home owner is fully aware of the status of the smart home subsystems and supervised spaces remotely through the internet using a computer or smartphone. The system notifies the user for emergency situations through sending email/SMS and sounding an alarm siren and/or speakers connected to the system.
- Logging of smart home system operations and historical data in electronic files.

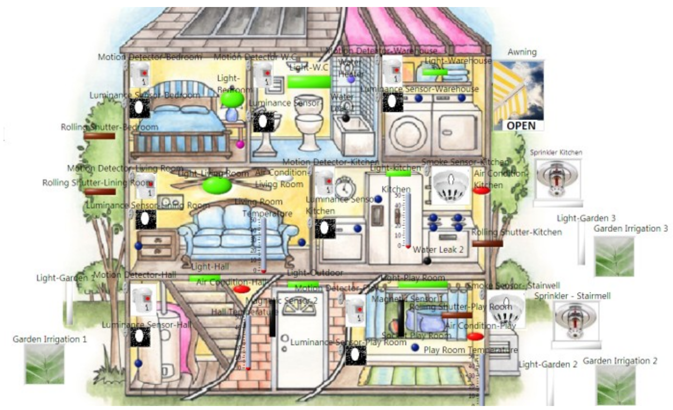


Figure 1. A demo smart home

Figure 1 illustrates a smart home mock-up depicting all measured and controlled parameters, sensors and devices. A similar mock-up represents the parking space linked to the house. Figure 2 illustrates the control panel of the implemented smart home, which also depicts a structured view of the supported smart home functionality grouped in various subsystems: alarms, lights, light indicators, parking, awnings (displayed in the left part of the console), security, air conditioning, rolling shutters, garden, water heater (displayed in the right part of the console), whiteboard and power budget (highlighted in the centre of the console).

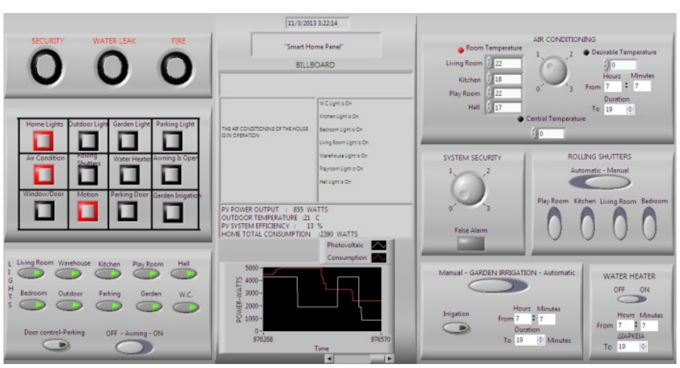


Figure 2. Smart Home Control Panel

A. Fire Safety

The requirement for home fire safety is self-evident. Nowadays, a constantly increasing number of houses are consuming an increased amount of electrical power. Therefore, active protection from fire is becoming a complicated process. Key objectives of fire protection are to ensure safety of human life and property protection, and to avoid fire expansion to neighboring houses or apartments in the block. Our system is employing 3 smoke sensors in fire risky areas: kitchen, stairwell and parking space. In case of fire, water sprinklers will throw water to extinguish it. The system makes decisions for enhanced fire safety, such as with opening blinds and windows, switching on lights at night or in case of low luminance, and opening the parking door in case of a fire in the parking space.

B. Security

The system employs magnetic traps in doors and windows, and motion detectors in all rooms. It features 3 operation modes: normal, with all security sensors active; sleep, with only the magnetic traps active; and panic, activated by the user. In all modes, an intrusion will activate the sirens, and lighting at night, and immediate notification of the owner.

C. Air Conditioning

The system meets different requirements for heating and cooling, depending on the use of each room. It offers 3 operation modes: “Room Temperature” will allow each room to have an own temperature; “Central Temperature” will set a common temperature across the home; the third mode integrates a timer in the second mode. Opening a window will signal the system to stop the air conditioning in the respective room.

D. Water Leak Detection

The system employs water leak detectors mounted on water pipes in the kitchen and bathroom. In case of water leak, the control system will switch off the water electrovalve to prevent water leak, and at the same time activate the siren and notify the owner. Furthermore, the event is recorded in the control panel switching on a led indication and posting a message on the whiteboard.

E. Awnings

Usually an awning needs to be closed in case of strong wind to prevent the damaging of it and causing further damages and accidents. The system monitors the wind speed and opens or closes the awnings accordingly. The control panel includes a button for manual operation of the awnings; however the system prevents the manual operation in case of strong wind.

F. Lighting

Out of the total home power consumption, 25% is due to lighting, while 4-5% out of this is due to unneeded lighting, either due to disuse or negligence despite a sufficient luminance level of ambient light. In order to avoid unnecessary power consumption, the control system automatically takes relevant decisions. Each room is equipped with a luminance sensor for measuring the level of ambient light, and a motion detector, the same one used by the security subsystem. Room lighting is activated only when the level of ambient natural light is insufficient and in parallel motion is detected in the room. The control panel includes an individual light button per separate space. The outdoor lighting is controlled via two parameters: the level of ambient natural light and a relevant button in the control panel. When the button is off, the external light is kept turned off even at night. When the button is on, the external light will turn on only when the ambient light goes low

G. Garden Irrigation

The control panel includes a button for selecting a manual or automated mode for garden irrigation as well as an irrigation button. In manual mode, pressing the irrigation button will start the irrigation process at anytime of the day. Irrigation stops when the relevant button is switched off. In automated mode, the user needs to define a start time and duration of the irrigation. The control system will shut down or not start the irrigation when it is raining.

H. Parking

The control system closes the door of the closed parking space when there is no motion detected. Door control is also accomplished via a relevant button in the control panel. The door will not close in case of a fire or motion in the parking space.

I. Water Heater

The control system allows setting the heating start time and duration of the water heater. If there is heated water available, heating will not activate.

J. Rolling Shutters, Blinds and Openning Systems

With setting the selection button in manual mode, each individual opening button defines the opening position of the corresponding rolling shutter or blinds. In automatic mode, the rolling shutters and blinds act automatically: they open during the day and close at night. In automatic mode, when a window and the corresponding rolling shutter are open and no motion is detected in the room for a long time, the control system infers that the window has been left open by the owner and will close the rolling shutter. In any mode, in case of fire, all rolling shutters will open automatically.

K. Energy Management

A modern house powered by renewable energy usually employs a photovoltaic installation. These PV systems are usually rooftop installations that collect solar power and convert it to electricity which can be used to power the house (selling the remaining energy to the grid). The rooftop PV system of our reference smart home has an 8KW total capacity. The power output depends on solar irradiance and ambient temperature. The control system receives those values from appropriate sensors, calculates the current power output and performance of the PV system, and displays in the control panel all the above mentioned parameters. The power output of the rooftop PV system is dynamically displayed in a graph against the current total home power consumption. The system is also able to log the information to enable several logistics and processing concerning home energy strategy.

The control system automatically applies the following rules to avoid unnecessary power consumption:

1. No light is switched on if the level of ambient light exceeds an upper threshold level.
2. Air conditioning in a room will be deactivated if a window is opened and remains open for some time.
3. The irrigation system is deactivated on a rainy weather.
4. Water heating will not activate if there is enough heated water available.

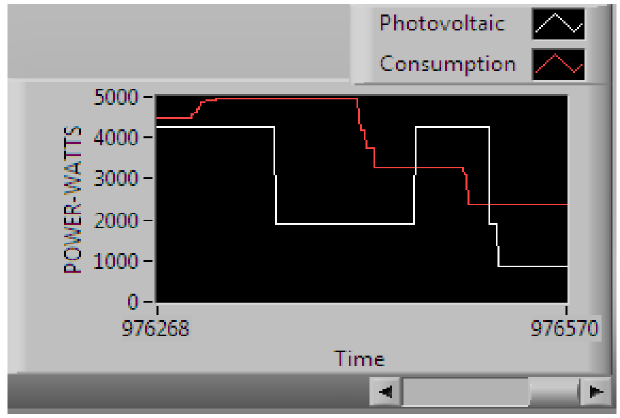


Figure 3. Home power generation and consumption vs. time.

III. LABVIEW DEVELOPMENT ENVIRONMENT

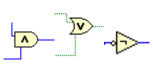
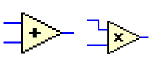
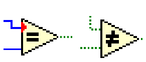
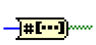
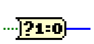
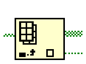
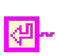
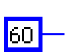
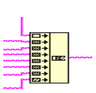
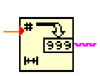
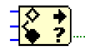
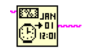
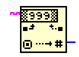
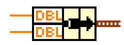
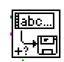
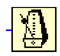
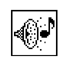
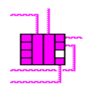
LabVIEW is a graphical programming environment suitable for building monitoring and control applications, including test and measurement, data acquisition, instrument control, datalogging, measurement analysis, and report generation. LabVIEW adopts a graphical code style called virtual instrumentation for building measurement and control systems involving hardware and software. A LabVIEW program is a Virtual Instrument (VI). Each virtual instrument comprises the front panel, which implements the graphical communication between the user and the LabVIEW environment, the block diagram, consisting of program execution nodes (functions, subVI programming structures and external routines), terminals and wires, and the connector, which defines the terminals used for connecting controls and indicators, as well as inputs and outputs between virtual instruments. Controls represent typical instrument input devices, such as knobs, push buttons and dials. Indicators represent instrument output devices and display data the VI acquires or generates, such as charts, thermometers and measurements. Functions are used for VI programming and include input/output instruments, files, data acquisition functions etc.

The interfacing of smart home sensors, actuators and devices is implemented through PC-based DAQ (data acquisition) devices available on a variety of PC buses. The role of a DAQ device is to acquire and measure analog or digital electrical signals from sensors, transducers, and fixtures and to generate analog or digital electrical signals. Furthermore, various external sensors could be used to convert a physical signal into an electrical unit, such as voltage or current, and vice versa. A DAQ icon is inserted in a LabVIEW program in order to interface a sensor, actuator or device with the control panel. A DAQ device should be properly configured through the LabVIEW DAQ Assistant as per the input/output and digital/analog characteristics, such as signal scale range, differential or referenced single ended (RSE) terminal mode, acquisition timing and sampling rate.

The main advantages of using LabVIEW for application programming lie among the significant ease of graphical programming, availability of rich library functions for data analysis and processing, data logging and exporting, ease of interfacing with external programs (via DLL, ActiveX etc.), networking with other applications and VIs, as well as web publishing to enable and control browser access to VI front panels.

Table 1 presents the virtual instruments which were used for the implementation of the interactive smart home application in order to better understand the description of the developed application in the ensuing section as well as to think of future application enhancements.

TABLE I. ANALYSIS OF SMART HOME VIRTUAL INSTRUMENTS

	Boolean control mirror
	Boolean indicator mirror
	String control mirror. It can accept 16-bit integers.
	String indicator mirror. It can display text or numbers depending on the wiring type
	Numeric control mirror. It can display 64–bit numbers with an accuracy of 15 digits
	True constant. It provides a true value in the program. Its value can be changed to false through the functions palette.
	Boolean functions AND, OR, NOT
	Adder/Multiplier of input numbers
	Equal/Not equal. They compare the input values and output a Boolean value (true or false)
	Number to boolean array. It converts an integer to a boolean array
	Boolean to (0,1). It converts a boolean value to a 16-bit integer (0 or 1)
	Index array. It extracts individual data elements from an array
	Case structure. Conditional branching control structure, that executes one of its subdiagrams based on the input to the case structure. It is the combination of the IF, THEN, ELSE, and CASE statements in control flow languages.
	Carriage return constant. It is used for changing a font type
	Numeric constant
	Concatenate strings
	Number to decimal string. It converts a number to a series of decimal digits. Floating numbers are rounded in a 64-bit integer before the conversion
	Range and coerce. Its output will be true or false depending on whether the measurable value is within the allowed limits, which are declared as parameters.
	Format date /time string. It displays time and date in the declared format
	Decimal string to number. It converts the numeric characters in the string to an integer and returns the number
	Bundle. It assembles a bundle of individual elements. It can change the values of only a subset of elements leaving the rest of the values intact.
	Charting waveform data
	File path
	Write characters to file. It is used for storing data in the defined file path
	Multiple. It decides the frequency of revisiting a program and is usually used for synchronization
	Beep. It makes a beep sound on a true input value
	Connector. It creates a connection with the data of another subVI

IV. DEVELOPMENT OF THE SMART HOME APPLICATION

The execution of a program in LabVIEW is based on the smooth flow of data. In other words, a node is executed only when all input data that previous nodes generate is available. In our case, the smart home application makes use of the sequence structure, which allows the repetitive execution of code segments individually and in a uniform logical order. In the ensuing, several representative program segments are presented in detail.

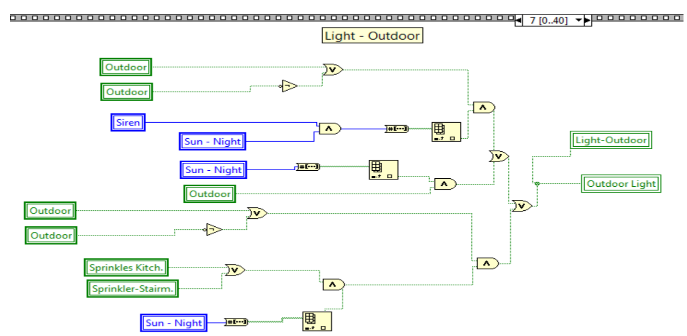


Figure 4. Condition for outdoor lighting and corresponding led indicator of the control panel.

Figure 4 displays the conditions defined for the activation of outdoor lighting and the corresponding control panel indication. Similar conditions are handling the entire lighting system. Each room has its own motion and luminance sensor and light button. The interconnections between elements of different data type are realized through the conversion of numerical data to boolean values.

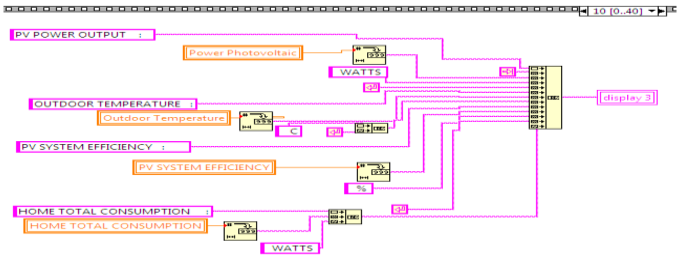


Figure 5. Display of power level and performance of the PV installation, outdoor temperature and total home power consumption

Figure 5 illustrates the display on the control panel of data involving the generated power level and performance of the PV installation, the outdoor temperature and the total home power consumption. Displaying is realized in a similar way across the system. Whenever the logical value of a terminal is accompanied with text, the case structure is used to convey the associated text in the bundle element, such as for displaying the home lighting status in the control panel announcements (see fig. 6).

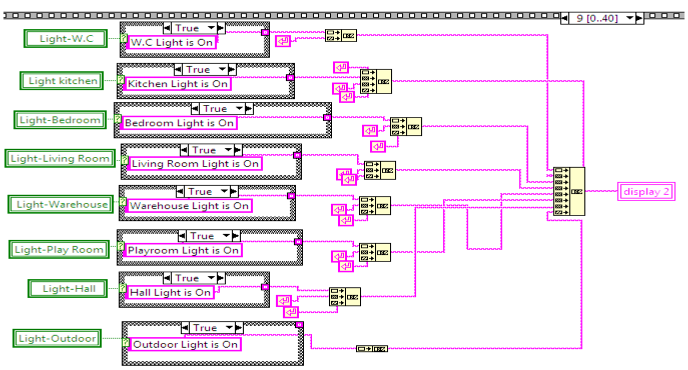


Figure 6. Display of home lighting status

Figure 7 illustrates the conditions referring to the automated watering. When the knob is set to automation mode the system executes the block diagram of the true case. In manual watering the system executes the block diagram of the false case. The new virtual instruments shown are used for clock insertion in the program and time conversion to minutes. A range structure is further shown, which outputs a logical 1 value when the input value is within the desirable range.

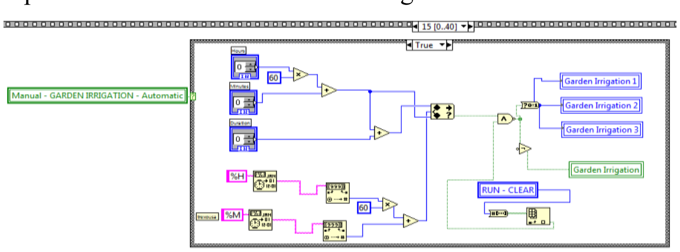


Figure 7. Condition for automated watering

Figure 8 illustrates the operation of the normal mode and the respective case program regarding the alarm unit. This is one of the four modes available via a knob.

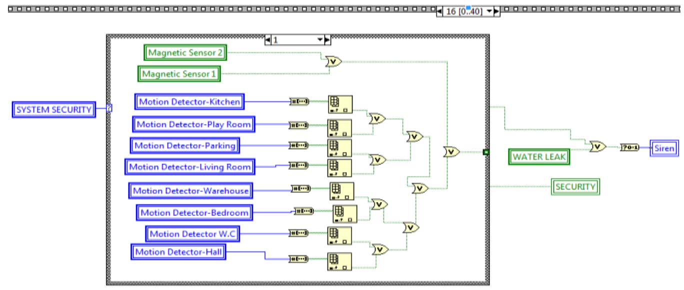


Figure 8. Alarm system in normal operation mode

Figure 9 illustrates the water heater operation mode, which is activated by a button or through a timer.

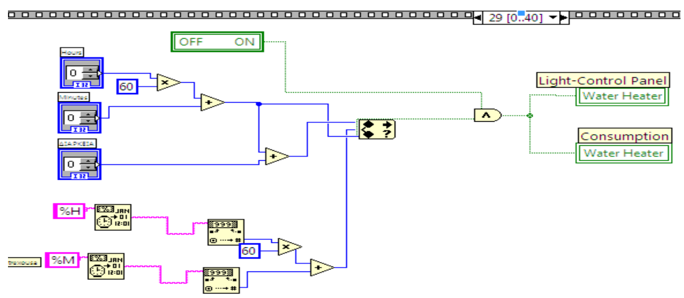


Figure 9. Condition for water heater control.

Figure 10 illustrates the operation of the third program mode of the air conditioning unit, which includes the timer setting.

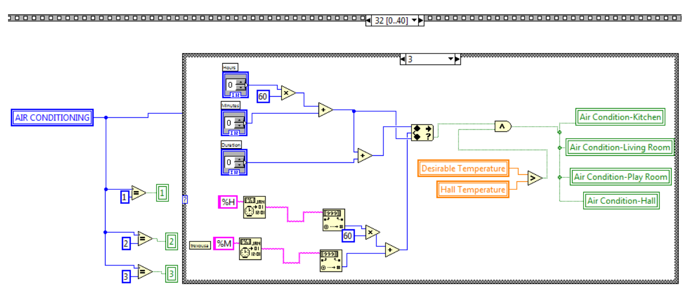


Figure 10. Activation of central air conditioning condition with timer

Figure 11 illustrates the calculation of the total home power consumption. A similar block diagram calculates the electrical power generated by the PV system (fig. 12). Both the generation and consumption of power drive a bundle which inserts the numerical values in a graph.

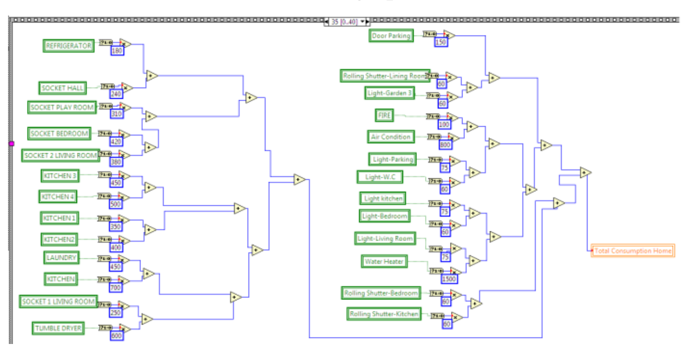


Figure 11. Condition for the calculation of home power consumption

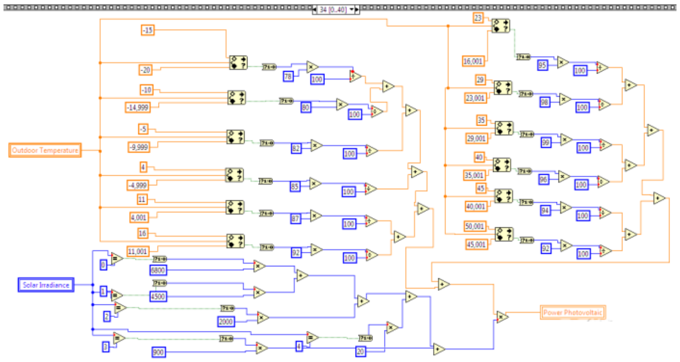


Figure 12. Condition for the calculation of rooftop PV electrical power generation depending on solar radiation and outdoor temperature.

In fig. 13 home power generation and consumption drive a bundle which inserts those data in a chart. This chart is presented in fig. 3. Figure 14 illustrates the code and the virtual instrument, inside the case structure, used to store the respective data in a user defined excel file.

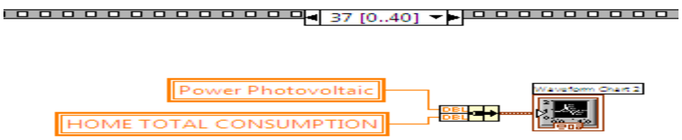


Figure 13. Power generation vs. consumption chart.

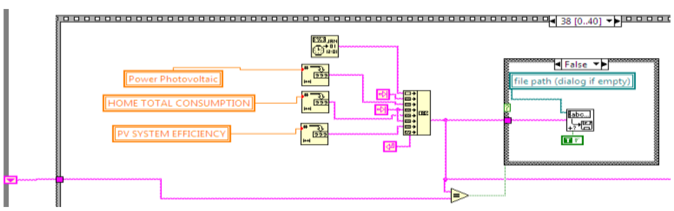


Figure 14. Condition for power related data storage.

Figure 15 illustrates the condition for email user notification to the home owners upon activation of the security or the fire alarm system. A predefined message and mailing parameters (outgoing mail server, gmail account authentication details, sender and recipient addresses) are declared in the subprogram, which further shows the gmail connector and an accompanying sound notification.

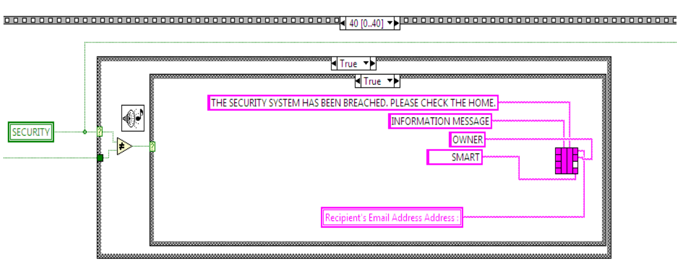


Figure 15. User notification through sending of e-mail.

V. Conclusions

We have developed a flexible modular smart home application using the visual programming paradigm and the LabVIEW environment, which meets successfully the initial user requirements for a completely functional system. LabVIEW is ideal for any measurement or control system, integrating graphical tools that facilitate the building of a wide range of applications in dramatically less time than using other practices. Evidently, the presented system modeling and implementation paradigm can be easily migrated to a broad range of automation applications.

The introduced system can be integrated perfectly and smoothly as an advanced front-end module with our proposed web-based three-tier architecture for collective large scale control and monitoring applications described in [9]. This collective system involves a service backend which undertakes to collect, store and manage data from remote installations, and provides in real-time user services for control, monitoring, notification, reporting and data export. Furthermore, the work presented here can easily contribute a consistent and interoperable smart home component to a next generation intelligent environment framework and ambient adaptive system, such as the one presented in [1], enabling applications that can pervade all levels of daily life to assist humans in their activities.

References

[1] T. Heinroth and W. Minker, "Next Generation Intelligent Environments: Ambient Adaptive Systems", Springer, Sept. 2011, ISBN 978-1-4614-1298-4.

[2] S. Helal, Programming Pervasive Spaces, IEEE Pervasive Computing, 4(1), 2005, pp. 84-87.

[3] National Instruments: Test, Measurements and Embedded Systems, LabVIEW System Design Software, http://www.ni.com/labview/

[4] S.-M. Baek, K.-B. Chang, I.-J. Shim, G.-T. Park, Implementation of smart home control using LabVIEW and PDA, IEEE Int'l Symposium on Consumer Electronics, Sept. 1-3, 2004, Reading, UK, pp. 558-562.

[5] S. Folea, D. Bordencea, C. Hotea, H. Valean, Smart home automation system using Wi-Fi low power devices, IEEE Int'l Conference on Automation Quality and Testing Robotics (AQTR), May 24-27, 2012, Cluj-Napoca, Romania, pp. 569-574.

[6] P. Janani, S. Indumathi, Energy Conservation in smart home using LabVIEW, Int'l Conference on Computing and Control Engineering (ICCCE 2012), Chennai, India, April 12-13, 2012.

[7] B. Hamed, Design & Implementation of Smart House Control Using LabVIEW, International Journal of Soft Computing and Engineering (IJSCE), 1(6), January 2012, pp. 98-106.

[8] N. Papadopoulos, A. Meliones, D. Economou, I. Karras, I.Liverezas, "A Connected Home Platform and Development Framework for Smart Home Control Applications", 7th IEEE Int'l Conference on Industrial Informatics (INDIN09), June 24-26, 2009, Cardiff UK.

[9] A. Meliones and A. Nouvaki, A Web-based Service Backend for Control and Monitoring of Solar Parks", 10th IEEE Int'l Conference on Industrial Informatics (INDIN2012), July 27-29, 2012, Beijing, China