International Journal of Computer Theory and Engineering, Vol. 3, No. 3, June 2011

Real-Time Application Surveillance Security System Based on LabVIEW

Basil Hamed

Abstract—**Nowadays surveillance security system is very important for companies and factories. No facility can operate without a surveillance security system. Remote real-time surveillance systems are built in a network which improves the power of application. The main purpose of this paper is to know the exact location and type of problem that may occur in the facility by monitoring the system using computers. This system can be relayed on to do the suitable reaction automatically when the facility is in real danger, and as an advanced option the supervisor can monitor and connect to the system at any time or place using the internet. This system consists of hardware and software tools, and the main tools are PIC (16F877), LabVIEW, Parallel Port and different type of sensors. In this paper, we developed a Real-time Online Inspection System to control VI remotely using LabVIEW**

***Index Terms*—Control, LabVIEW, surveillance security, and PIC16F877A**

I. Introduction

The protection of people and valuable assets against loss is one of the primary concerns for both private and public sector businesses. If we currently have or intend to install security, fire protection, detection or evacuation systems to protect people and assets, we must select a perfect system which prevents destruction of life, property and institutions. For a complete system, monitoring is needed to see actions online since the alert is on, LAN access is available for users, records and actions archive will be necessary for proceeding and controlling, cameras will be useful in the system. For more safety, an automatic reaction must be done if there was any high level alert. Remote real-time surveillance systems are built in a network which improves the power of application [1].

The system contains three major parts: the main board sections, the secondary board section, and the monitoring and data storing section. We used in the secondary and the main section (board) PIC (16f877), but form monitoring and storing we have used LabVIEW, and Parallel Port used to connect the main PIC and LabVIEW.

The surveillance security system will use different type of sensors such as; fire detectors, motion and security sensors. These sensors are connected to a PIC16F877 for processing by transforming signal into codes and transmit it to a main board. This signal will be responsible of taking automatic reaction and transmitting codes to the monitoring program. The monitoring program will be responsible of receiving codes and converting the codes to actions and indicators to be fixable with user interface, also it will preview a video signal of monitoring cameras. Network monitoring and controlling will access more flexibility on the system, web browsing and employees database is added to give more securing to monitoring and to prevent breaking into the facility.

Manuscript received October 8, 2010; revised January 4, 2011.

Basil Hamed is with Islamic University of Gaza, Palestine (email: bahamed@hotmail.com).

II. Hardware Components

The hardware components used in the surveillance security system are; PIC16F877 Microcontrollers, Parallel Port, LCDs, Keypad and Sensors

A. *PIC16F877A*

A. PIC16F877A as shown in Figure (1), it has a set of serial ports built in, which are used to transfer data to and from other devices, as well as analogue inputs, which allow measurement of inputs such as temperature. PIC16F877A is used widely as a more advanced teaching device, because it has a full complement of interfaces: analogue input, serial ports, slave port and so on, plus a good range of hardware timers [2].

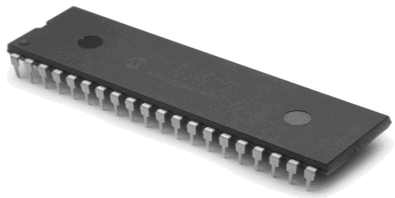


Figure (1) PIC16F877A

PIC16F877 has a total of 40 pins. The chip can be obtained in different packages, such as conventional 40-pin DIP (Dual In-Line Package), square surface mount or socket format. Most of the pins are for input and output, and arranged as 5 ports: A (5), B (8), C (8), D (8) and E (3), giving a total of 32 I/O pins. These can all operate as simple digital I/O pins, but most have more than one function, and the mode of operation of each is selected by initializing various control registers within the chip. Note, in particular, that Port A and port E become analogue inputs by default (on power up or reset), so they have to set up for digital I/O if required. Port B is used for downloading the program to the chip flash ROM (RB6 and RB7), and RB0 and RB4–RB7 can generate an interrupt. Port C gives access to timers and serial ports, while Port D can be used as a slave port, with Port E providing the control pins for this function [3].

B. *Parallel Port*

Traditionally IBM PC systems have allocated their first two parallel ports according to the configuration as shown in Table (1)

TABLE (1): PARALLEL PORT RESOURCES

Printer	Data Port	Status	Control
LPT1	0378h	0379h	037ah
LPT2	0278h	0279h	027ah

The registers that are located at the addresses in Table (1) are listed in table (2):

TABLE (2): PARALLEL PORT REGISTERS.

Addres s	Description	Register Type	Read/Write
Base + 0	LPT Data	Register Data	R/W
Base + 1	LPT Status	Register Status	R
Base + 2	LPT Control	Register Control	W

Notice that the registers shown in the table (2) are all of different types. Since data can flow both ways on the parallel port, the LPT Data Register is both R/W. The LPT Status Register says whether or not voltages have been applied to the status pins.

C. *LCDs*

Liquid crystal display or LCD is a very commonly used device in electronics projects to display data and interact with users. To use LCD effectively in microcontrollers, some initial experiments can performed by connecting up a series of switches to the pins of the module as presented in Figure (2).

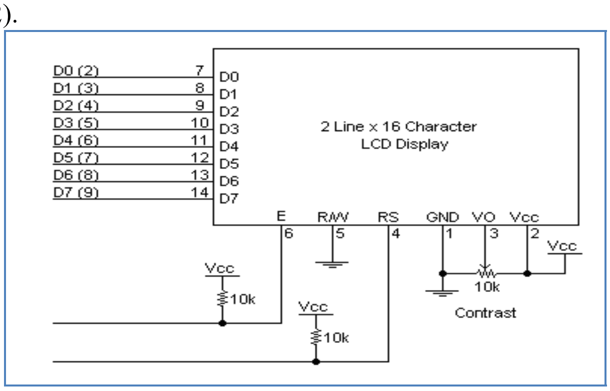


Figure (2): Circuit diagram for LCD Connections

D. *Matrix Keypad*

Keypad is a commonly used device to get the user input. Although simple push switches can be used to get user input, as has been done for this system, this would require 1 I/O line per switch.

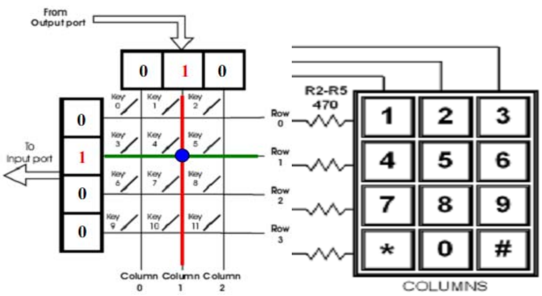


Figure (3): Keypad rows and columns

The keypads Shown in Figure (3), has three columns and 4 rows.

III. SOFTWARE

The software's used in the surveillance security system are: LabVIEW, MikroC, Proteus 7 Professional, LabSQL ADO, and MS FrontPage

A. *LabVIEW*

LabVIEW is a software and registered trademark of National Instruments. It is a popular visual programming environment for data acquisition, analysis, and control. A visual programming interface allows the user to create programs by connecting objects with various functions in a flow diagram, such as DAQ VIs, buttons, and graphs.

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where data determine execution.

In LabVIEW, the user builds a user interface by using a set of tools and objects. The user interface is known as the **front panel**. The code, contains in the **block diagram,** is a graphical representation of functions controls the front panel objects. [4]

Front panel

The front panel is the user interface of the VI. You build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays .Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates [5].

Block diagram

After building the front panel, add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. It cannot delete a terminal from the block diagram. Every control or indicator on the front panel has a corresponding terminal on the block diagram. Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries.

B. *MikroC*

MikroC is a powerful, feature rich development tool for PIC micros. It is designed to provide the programmer with the easiest possible solution for developing applications for embedded systems, without compromising performance or control. [2]

C. *Proteus 7 Professional*

Proteus 7 is an interactive system level simulator. Which combines mixed mode circuit simulation, micro-processor models and interactive component models to allow the simulation of complete micro-controller based designs. [6]

D. *MS FrontPage*

FrontPage 2003 provides the features, flexibility, and functionality to help build better Web sites [7]. Take advantage of professional design, authoring, data, publishing tools and everything that needed to create a dynamic, sophisticated Web presence.

E. *ODBC*

Open Database Connectivity (ODBC) is a technology programs use to access a wide range of databases (or data sources) [8]. In our system, data need to export from LabVIEW into an Access database. To do this, the correct ODBC driver and data source needed.

F. *MS Access*

Microsoft Access is a powerful program to create and manage databases [9]. It has many built in features to assist in constructing and viewing data. Access is much more involved and is a more genuine database application than other programs such as Microsoft Works.

G. *LabSQL ADO functions*

Microsoft® ActiveX® Data Objects (ADO) enables the user to write an application to access and manipulate data in a database server through an OLE DB provider. Its primary benefits are ease of use, high speed, low memory overhead, and a small disk footprint. ADO supports key features for building client/server and Web-based applications [10]. ADO also features Remote Data Service (RDS), by which we can move data from a server to a client application or Web page, manipulate the data on the client, and return updates to the server in a single round trip. Previously released as Microsoft Remote Data Service 1.5, RDS has been combined with the ADO programming model to simplify client-side data remote.

IV. DESIGN

This section will explain the design procedure and how the surveillance security system works. In general, sensors will detect the actions and indicate the secondary PIC. A code will transmit to the primary PIC which will connect the LabVIEW designed program, and then the computer will save the actions in the database and take a reaction as shown in Figure (4) and Figure (5).

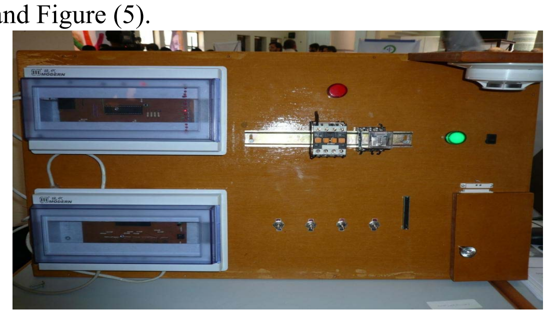


Figure (4): System Hardware

The primary PIC will receive a code consist of two packages, every package contains 9 bits, the first bit is "1" that means starting transmitting; the other 8 bits are for sensors status. The transmitted code has the form of "100000000100000000" which will be generate from the secondary PIC. The primary PIC will process the code and follow it with another 3 bits containing the sensor type indication or the location of the action. Then the primary PIC will transmit the code for the LabVIEW using parallel port. This code must be saved in the database to be dealt with; a comparison is made to know the action type and the place where it occurs. All of these actions will be stored in a database, and user will be able to see these actions and control the system from the network. Now discussion of these procedures is presented.

Any indication on any sensor will make interrupt to a secondary PIC which will transmit bit to the main PIC to start receiving the coming message. A master slave protocol is used which will generate a 16 pulse to control received bits from secondary PIC. When the main PIC receives any data package it will send them to LabVIEW with the code of the branch where the interrupt occur in the same way of master slave method .The packages will be processed in LabVIEW to indicate the exact action which happened in the system.

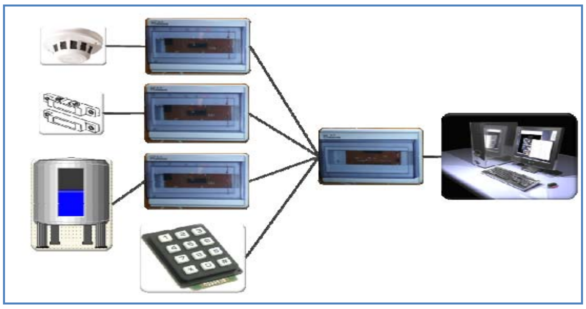
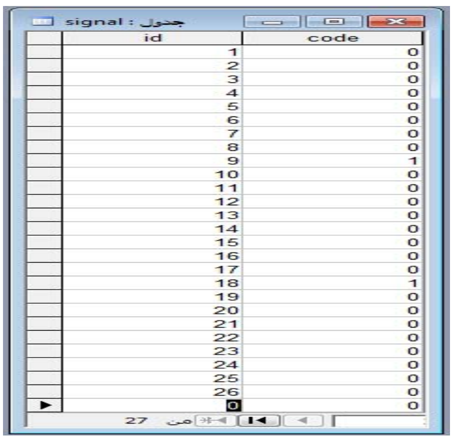


Figure (5): Connection Diagram

A. *LabVIEW*

Record Coming Signal

In-Port.vi 379h is responsible of receiving codes from parallel port; data is transmitted from primary PIC on the S3 pin by parallel port. If any "1" bit is received, the LabVIEW will transmit a 25 pulses which are transmission protocol. The received code will be updated in the signal table of the database. See Figure (6) and Figure (28) the SQL ADO execution statement which execute the database exportation is "UPDATE signal SET code = 0 WHERE id= row no" .Where signal is the table name and code is the column name.



signal : جدول

id	code
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	1
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	1
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
0	0

27 من

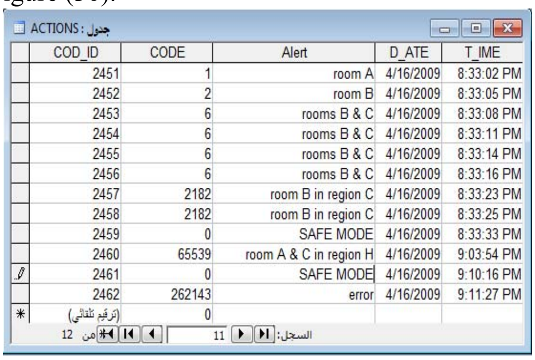
Figure (6): Signal Table in Database

Read Code from Signal

In this block (Figure 29), LabVIEW will read the stored bits from the signal table and then will masking it to omit unwanted bits, the statement of " SELECT code FROM signal ORDER BY id" in the execute ADO is responsible of importing code data from database. The code is 26 bit separated to three packages, the first and second packages are data information and it needs the first bit of these packages only to transmit the code so it should get rid of them. The first three bits of the third package (the branch code) is needed so we should get rid of the other five bits. The result will be arranged in 16 bit represents the sensors status in cluster form and 3 bits of the region in an array form.

Record Actions

In this block we will compare the new and the last action to avoid repetition in action records. If there is no repetition in codes or test mode is off, a new record will be inserted. Otherwise it will not insert any records. The SQL ADO will query about the last CODE recorded in the ACTIONS Table using this statement “SELECT CODE FROM ACTIONS ORDER BY COD_ID DESC” and the result will be compared with the received code from signal Table. When a new action happened; the SQL ADO will insert a new record in the ACTIONS Table. Time and date of the action are important for actions query, so we have to add time and date in the ACTIONS Table, the SQL statement which execute recording is " INSERT INTO ACTIONS (CODE, ALERT_1, D_ATE,T_IME) VALUES (action code, alert description, 'date’,’ time')". Also it needs the alert notification to be clear so it can be known where the action happened, See Figure (7) and Figure (30).


ACTIONS : جدول

	COD_ID	CODE	Alert	D_ATE	T_IME
	2451	1	room A	4/16/2009	8:33:02 PM
	2452	2	room B	4/16/2009	8:33:05 PM
	2453	6	rooms B & C	4/16/2009	8:33:08 PM
	2454	6	rooms B & C	4/16/2009	8:33:11 PM
	2455	6	rooms B & C	4/16/2009	8:33:14 PM
	2456	6	rooms B & C	4/16/2009	8:33:16 PM
	2457	2182	room B in region C	4/16/2009	8:33:23 PM
	2458	2182	room B in region C	4/16/2009	8:33:25 PM
	2459	0	SAFE MODE	4/16/2009	8:33:33 PM
	2460	65539	room A & C in region H	4/16/2009	9:03:54 PM
	2461	0	SAFE MODE	4/16/2009	9:10:16 PM
	2462	262143	error	4/16/2009	9:11:27 PM
*	(ترقيم تلقائي)	0			

السجل: 11 من 12

Figure (7): ACTIONS Table in Database

Alert Query

The block in Figure (32) is responsible of finding the exact alert type with the same code received, we have made a table which contains all available alert descriptions in Figure (31), when any action occurs it will use the query function to find its description then display it on the alert window. The SQL ADO query statement is " SELECT DESC (branch No.) FROM SEN_DESC ". If any two actions or more happened in the same time, the alert must contains the description of all actions. So we have query about all sensors descriptions and enable the exact description.

Test Mode

Test mode is an important process to test the system if it is working properly. In test mode a Boolean 1 must get out to the main PIC to inter the system in testing mode, also it will disable Record Actions stage to separate true actions and virtual actions. The Out-Port 378h can be used to provide the system with a DC 5 V as seen in Figure (8).

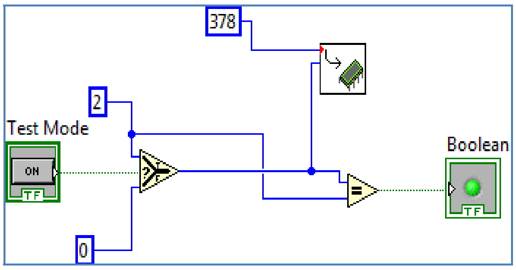


Figure (8): Test Mode Enable

The Main Board

The main function of the PIC (16f877a) is to manage a connection between sensors and the VI LabVIEW. It is responsible of receiving data from the secondary PIC and distributes it on branches’ codes, after that it will send data to LabVIEW. A transmission protocol must considering synchronization, response and number of bits to be sent.

B. The Main Board

The main function of the PIC (16f877a) is to manage a connection between sensors and LabVIEW. It is responsible of receiving data from the secondary PIC and distributes it on branches’ codes, after that it will send data to LabVIEW. A transmission protocol must considering synchronization, response and number of bits to be sent.

Input/output Units

To use the most pins of the PIC 16F877A the number of sensors which are needed for surveillance security system must be known. We can add 8 regions and there components, so initialization the inputs and the outputs ports must be done.

```
void main() {
TRISb = 0b01010101;
portb=0x00;
TRISd = 0b01010101;
portd=0x00;
TRISc = 0b00000010;
portc = 0x00;
```

Delay

Delay is used to make a period between executing transmission and receiving data from ports, this delay makes stability in processing the coming and leaving codes.

```
delay_ms(3000);
```

Received Data Tests

The system must be stand by, waiting for any changing in the received data, a while loop is used to supervised a changing in received data.

```
while (1) {
```

Data received from one of the ports (region) is checked by

```
if (portb.f0==1){
```

After recognizing the region, a code of the region is added.

```
Pic_Num=1;
```

The next code can indicate data changes.

```
Test_DataPin.f0=1;
```

If change occurs; the PIC must be ready to be received.

```
Recive_Data_1();
Recive_Data_2();
```

Data must be transmitted to LabVIEW.

```
send_data();
```

This code will be repeated 8 times to cover all regions.

C. The Secondary PIC

Different sensors types are used to indicate actions (digital, analog or a keypad)

Digital / Analog Sensors

For digital sensors a 16 pins are initialized as inputs and 8 pins for analog sensors

```
void main() {
TRISb = 0b11111111;
TRISd = 0b11111111;
ADCON1 = 0x80;
TRISA = 0xFF;
TRISE = 0b00000010;
porte=0;
TRISc = 0b01010110;
portc = 0x00;
```

A delay is needed to save electricity stability

```
delay_ms(5000);
```

The system must be stand by, waiting for any change in the sensor status, a while loop is used to supervise the change in the sensor status.

```
while (1) {
```

In the next code, the new status of the sensor is compared with the last status and gives a reading

```
Compare();
```

If the result of comparing was not equal zero then the PIC will record the reading of the sensors in a register

```
ReadInputs();
```

Test Mode

If the PIC received a signal of testing mode from LabVIEW it will energize all the sensors to make indicates to PIC then transmit the generated code to the main PIC.

```
if (portc.f2==1){
sensgroup1_000
=0b11111111;
sensgroup2_000
=0b11111111;
}
```

This code is testing any changing in the input status

```
TestInputs();
```

The next code is needed to determine the indication is in digital or analog sensors; also it is responsible of transmission protocol.

```
if (Test_DataPin == 1){
Send_Data();
}
if (Test_DataPin_Analog ==
1){
Analog=1;
Send_Data();
Analog=0;
}
if(Test_DataPin_Security==1){
Analog=2;
Send_Data();
Analog=0;
```

D. Keypad

Keypad device works via login system which connected with the main board. Users' data will be transmitted to LabVIEW to be recorded as an action. First initialize input output ports and pins, and then initialize the LCD monitor. This will go into loops waiting for input and when the user enters his own password it will make some processes and print out *password correct* or *password wrong* on LCD and sends a signal to make suitable reaction, a code will be transmitted to LabVIEW to record the action.

```
void main() {
TRISd = 0b00000010;
trisc = 0b01111000;
trisa = 0b00010000;
portd = 0;
while(1) {
perm=0;
pass=0;
user=0;
pass_error=0;
No_Error=0;
Typing=1;
if(Typing==1){
Lcd_Custom_Config(&PORTB,7,6,5,4,&PORTB,2,0,3);
// Initialize LCD on
PORTB
Lcd_Custom_Cmd(Lcd_CURSOR_OFF); // Turn off
cursor
Lcd_Custom_Out(1, 1, text_0); // Print text at LCD
}
Test_pass();
if(perm==1){
porta.f0=1;
delay_ms(2500);
porta.f0=0;
Send_Data();
}
if(perm==2){
Send_Data();
}
}
}
```

E. Remote User Controlling

As a complete system, users must be connected with the surroundings. Also administrators can watch the system anytime and anywhere. Employee logs must be recorded to make more security. It is decided to make network access to LabVIEW available, and to do that must have a server and a site which will tide the relation and distribute the permissions on users.

LabVIEW Web Server

The Web server allows users to view the front panels of applications from a remote machine using a browser [11]. Both static and dynamic images of a front panel VI can be viewed remotely. Not only will the Web server allow us to view the front panel of an application, but we have the ability to control the VI as well. We can control the application or front panel remotely using a browser. To activate the web server, choose options from the tools menu and select web server configuration as seen in Figure (9).

Check Enable Web Server and press OK.

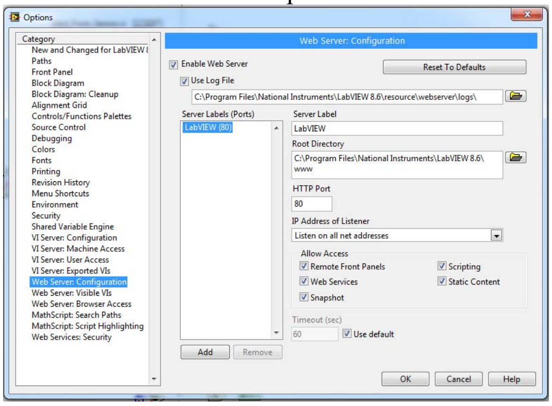


Figure (9): LabVIEW Web Server Options

Figure (10): Login Window

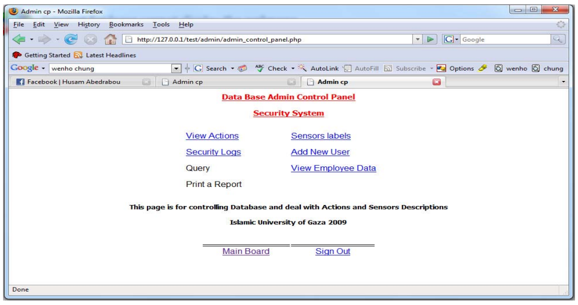


Figure (11): Admin Control Panel

Administrator Control Panel

As an administrator a user name and password must be available to secure the system as shown in Figure (10). The administrator is the only user who has the whole control and access to the system; the administrator can add users and employees edit their data and accounts. Also he can see all the actions in the data base and deal with it. Sensors descriptions are included to be edited and managed as the system required. For employees' security of arriving the system provides logs record to check the status of the employees Figure (11).

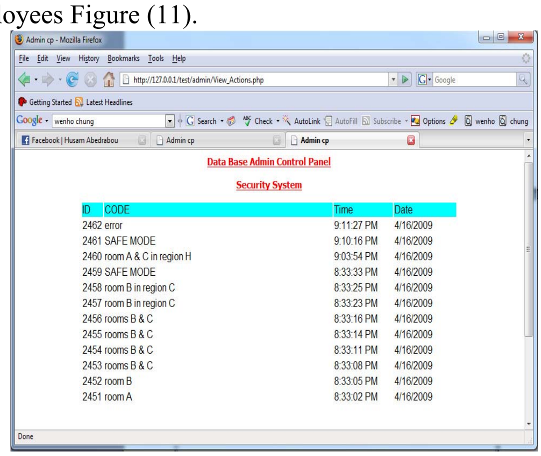


Figure (12): View Actions in Database

View Actions

Administrator or users with permission to access this page can read all actions happened in the system when the system is in work mode (Figure (12)).

Sensors Labels

To make alerts appears with the right indication of the sensors the system must be flexible to the changes of sensors places in the system, See Figure (13) and Figure (14).

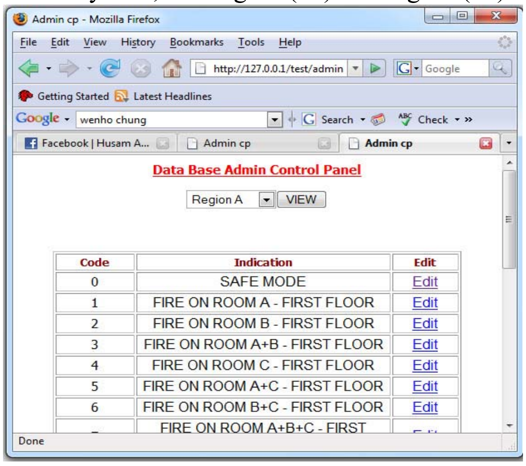


Figure (13): Sensors Codes and Descriptions

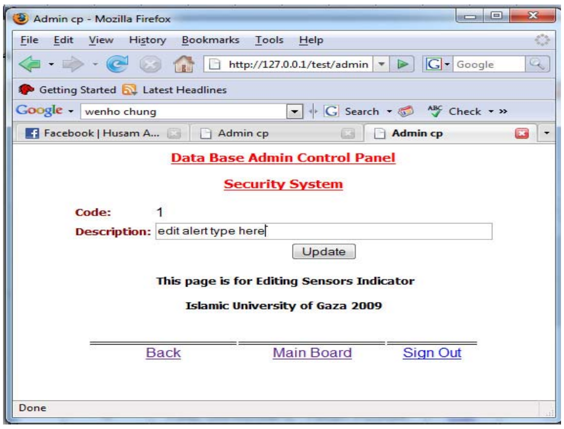


Figure (14): Edit Sensor Description

Security Logs

Security log allows administrator to see the movements of the employee in the company (arriving time and leaving time). The next PHP code is to display employees' names from database as shown in Figure (15).

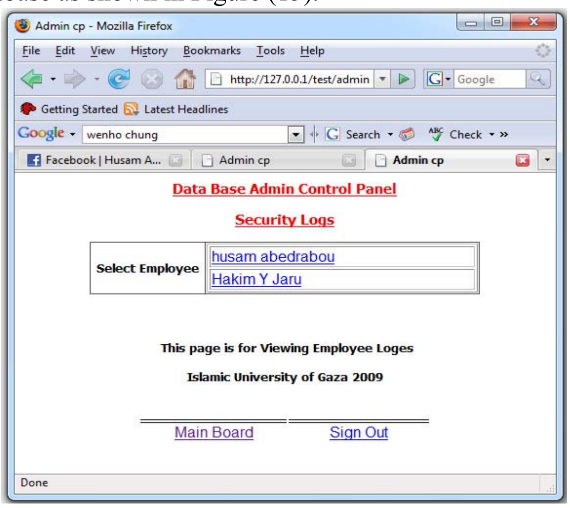


Figure (15): Employees Names in Database

```
<? $query = "select * from users ";
$result = odbc_exec($link, $query);
if($row = odbc_fetch_row($result)){
do {
$id = odbc_result($result,"user_id");
$Name = odbc_result($result,"Name"); ?>
```

The next PHP code is to display employees' logs from database as shown in Figure (16).

```
$myquery1 = "select * from come where
user_id like '$id'";
result1 = odbc_exec($link, $myquery1);
if($row = odbc_fetch_row($result1)){
do {
$date = odbc_result($result1,"dat");
$time1 = odbc_result($result1,"tim1")
```

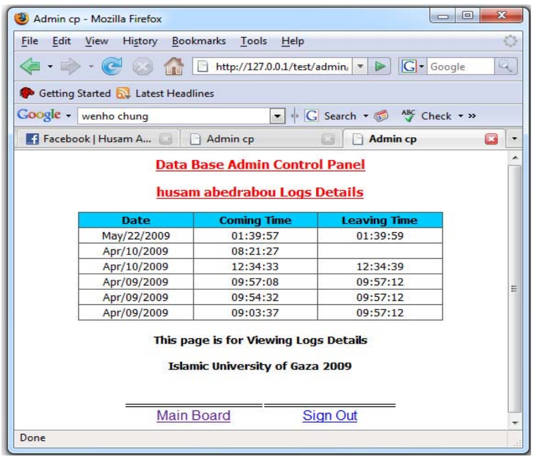


Figure (16): Employees Logs in Database

Add New User

When the system is new or a new user attends the company, a new account must be made for the new user. This new account gives the new user permission to access control or monitoring panel and make an archive of employee data as shown in Figure (17).

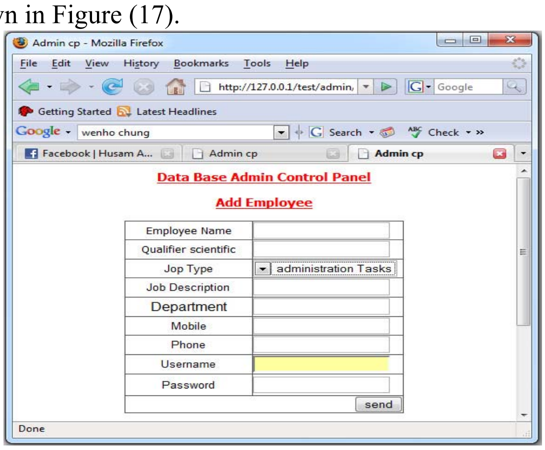


Figure (17): Add New Employee Form

View Employee Data

In this window the administrator can view employee's data, edit or delete as shown in Figure (18).

```
$myquery = "select * from users";
$result = odbc_exec($link, $myquery);
if($row = odbc_fetch_row($result)){
do {
$id = odbc_result($result,"user_id");
$Name = odbc_result($result,"Name");
```

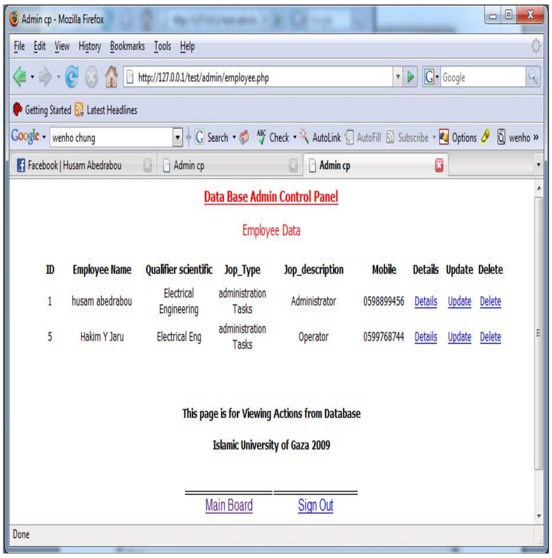


Figure (18): Employees Data Viewer

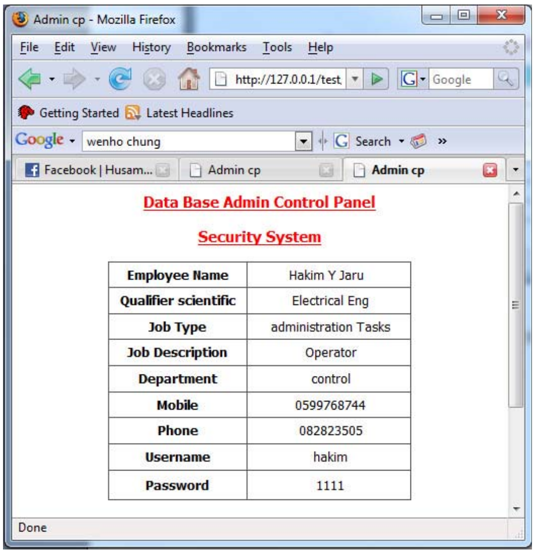


Figure (19): Employee Data View

Report Printing

As a complete system reports are necessary to be added in the archive as a hard copy.

Employee Control Panel

Users and employee need accounts to access the system for monitoring or controlling the VI LabVIEW. See Figure (20).

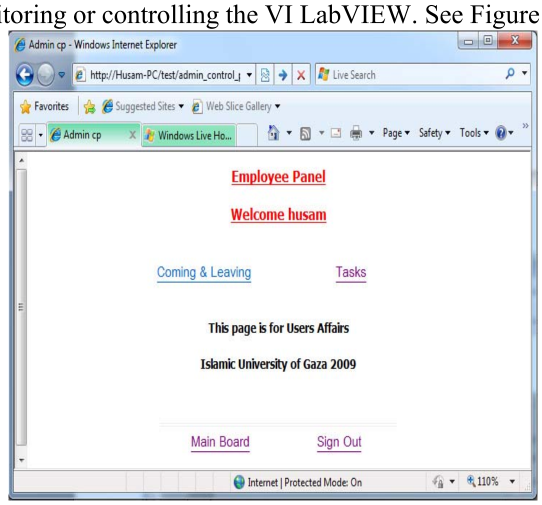


Figure (20): Employee Panel

Coming & Leaving

The window shown in Figure (21) is designed to record logs and be in touch with administrator in the same time.

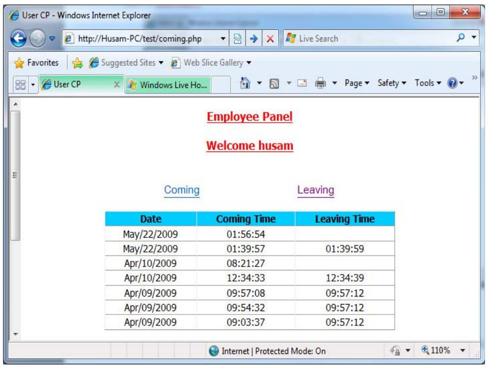


Figure (21): Employee Logs Recorder

Monitoring & Controlling

Every user will have an account in his name and special permissions of monitoring or controlling over network is granted as shown in Figure (22).

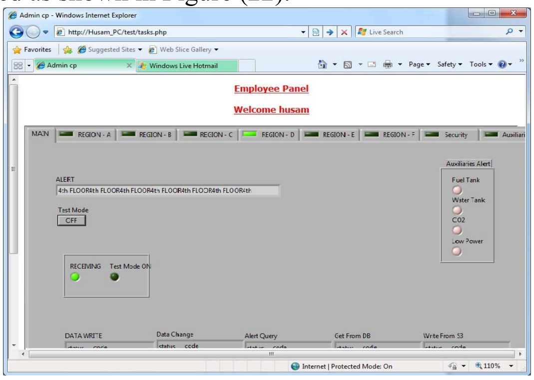


Figure (22): Monitoring & Control Over Network

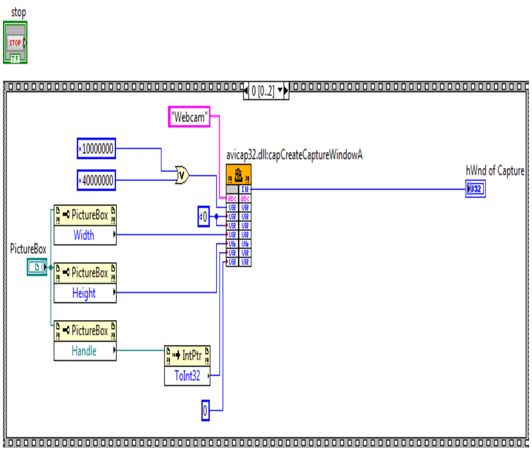


Figure (23): Video Monitoring

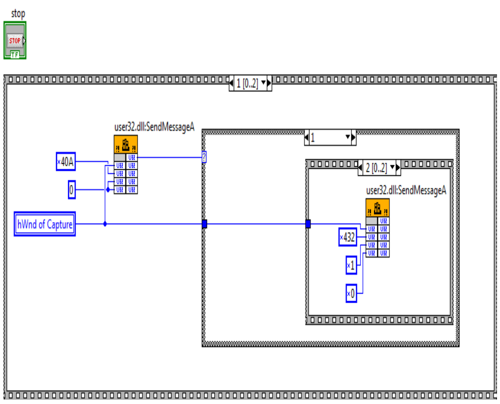


Figure (24): USB Web Camera Viewer

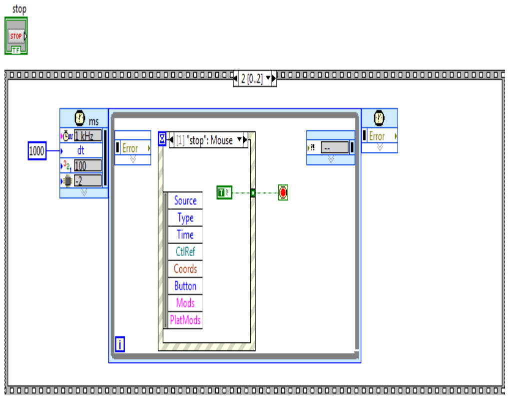


Figure (25): Video Sampling and Capturing

Video Monitoring

For video monitoring a web camera have been used with ActiveX object. The simple VI program to run USB Camera is explained in Figure (23), Figure (24) and Figure (25).

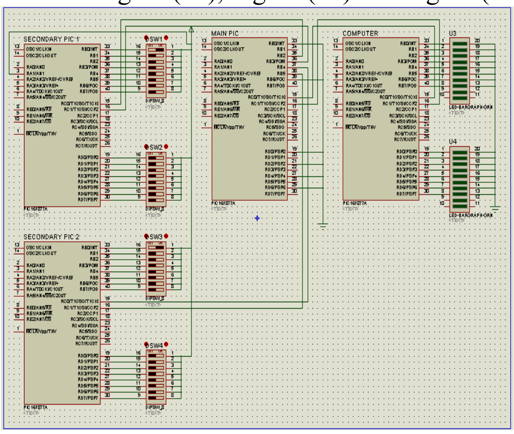


Figure (26): Main Board Circuit Diagram

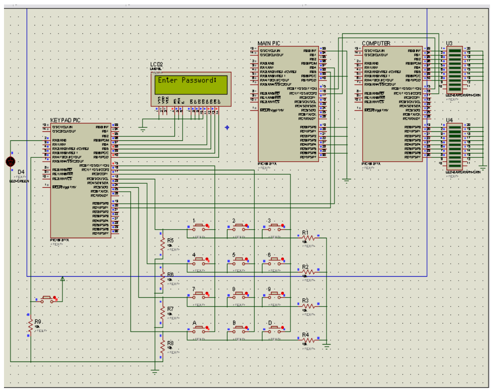


Figure (27): Secondary Board Circuit Diagram

Circuits Diagrams

A circuit diagram is a simplified conventional pictorial representation of an electrical circuit. It shows the components of the circuit as simplified standard symbols, and the power and signal connections between the devices. Arrangement of the components interconnections on the diagram does not correspond to their physical locations in the finished device.

Main and Secondary Board Circuit Diagram

Figure (26) provide a simulation program to test how the surveillance security system works in a virtual computer. Proteus 7 program has been used for this test.

Keypad Circuit Diagram

Figure (27) shows Proteus 7 a simulation program to test how the keypad is working on the program.

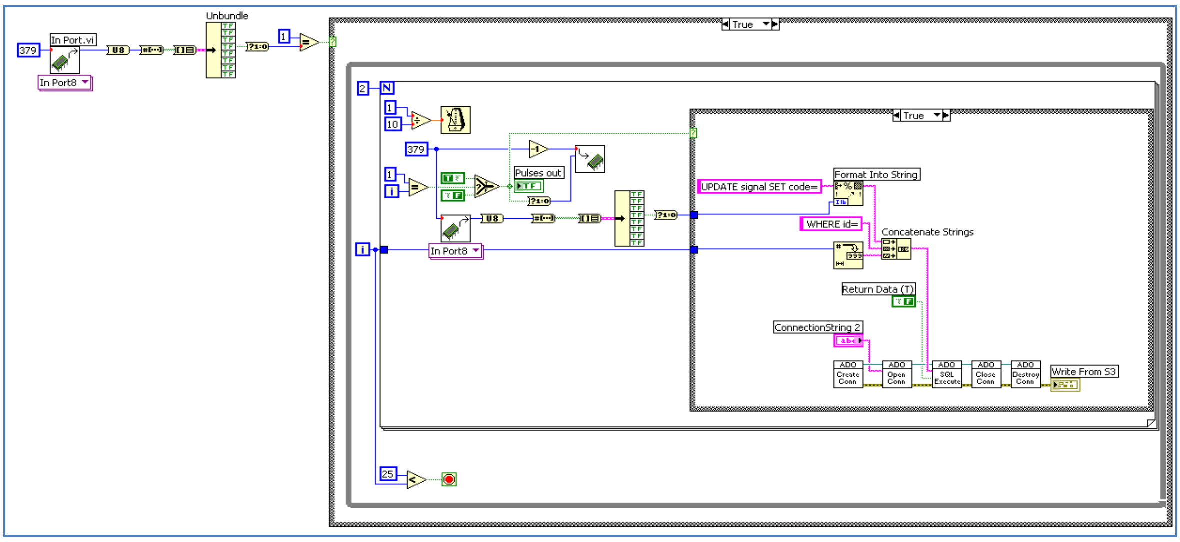


Figure (28): Read from Parallel Port

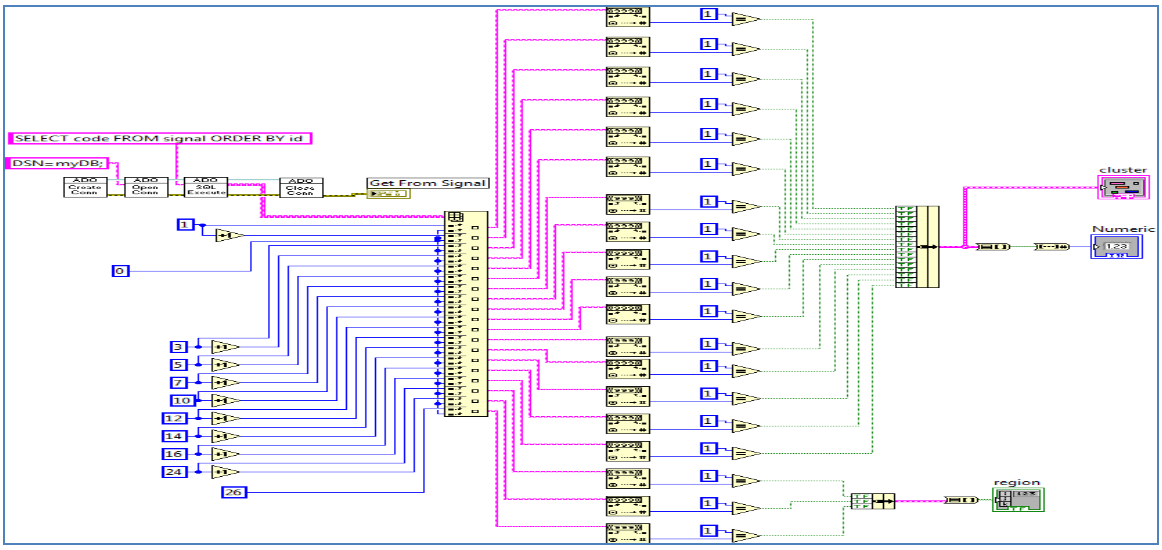


Figure (29): Read from Signal Table in Database

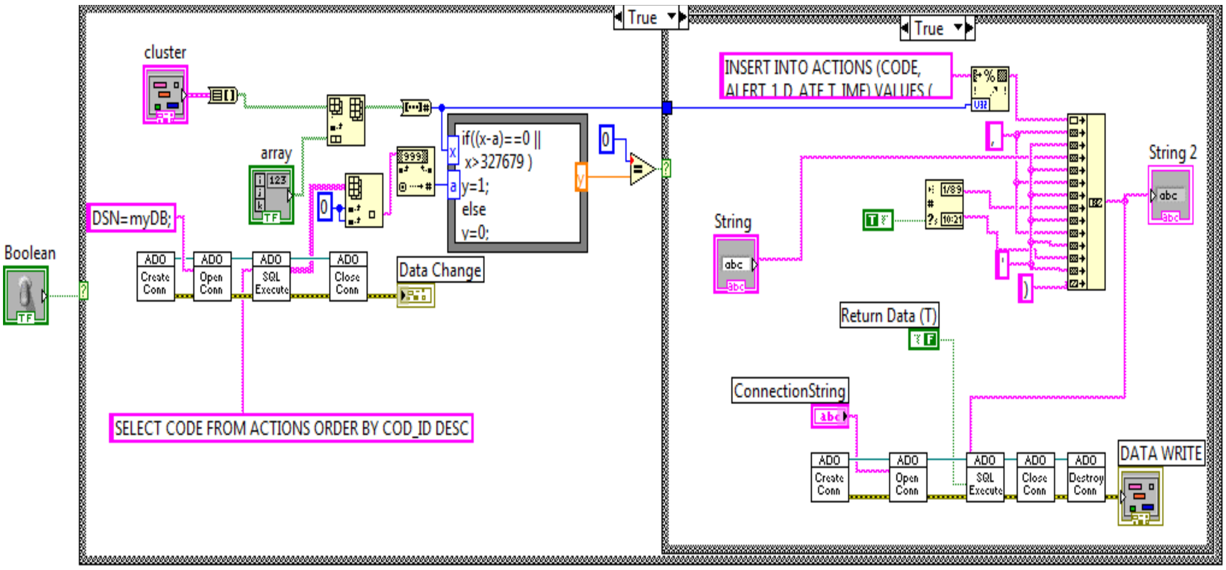


Figure (30): Record Actions

SEN_DESC : جدول

	ID	CODE	DESC0	DESC1	DESC2	DESC3	DESC4	DESC5	DESC6	DESC7
▶	1	0	SAFE MODE	SAFE MODE	SAFE MODE	SAFE MODE	SAFE MODE	SAFE MODE	SAFE MODE	No Change
	2	1	FIRE ON ROOM A - FIRST FLOOR	A - 2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 100 is open	0% auxiliary water
	3	2	FIRE ON ROOM B - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 101 is open	50% auxiliary water
	4	3	FIRE ON ROOM A+B - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 102 is open	100% auxiliary water
	5	4	FIRE ON ROOM C - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 103 is open	0% fuel
	6	5	FIRE ON ROOM A+C - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 104 is open	50% fuel
	7	6	FIRE ON ROOM B+C - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 105 is open	100% fuel
	8	7	:E ON ROOM A+B+C - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 106 is open	poor batarypower
	9	8	moderator	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 107 is open	good batary power
	10	9	FIRE ON ROOM D+A - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 108 is open	100% patary power
	11	10	FIRE ON ROOM D+B - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 109 is open	
	12	11	:E ON ROOM D+B+A - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 110 is open	
	13	12	FIRE ON ROOM D+C - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 111 is open	
	14	13	:E ON ROOM D+C+A - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 112 is open	
	15	14	:E ON ROOM D+C+B - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 113 is open	
	16	15	ON ROOM D+C+B+A - FIRST FLOOR	2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 114 is open	
	34	16		2nd FLOOR	3rd FLOOR	4th FLOOR	5th FLOOR	6th FLOOR	door 115 is open	

السجل: 1 من 17

Figure (31): Sensors Description Table in Database

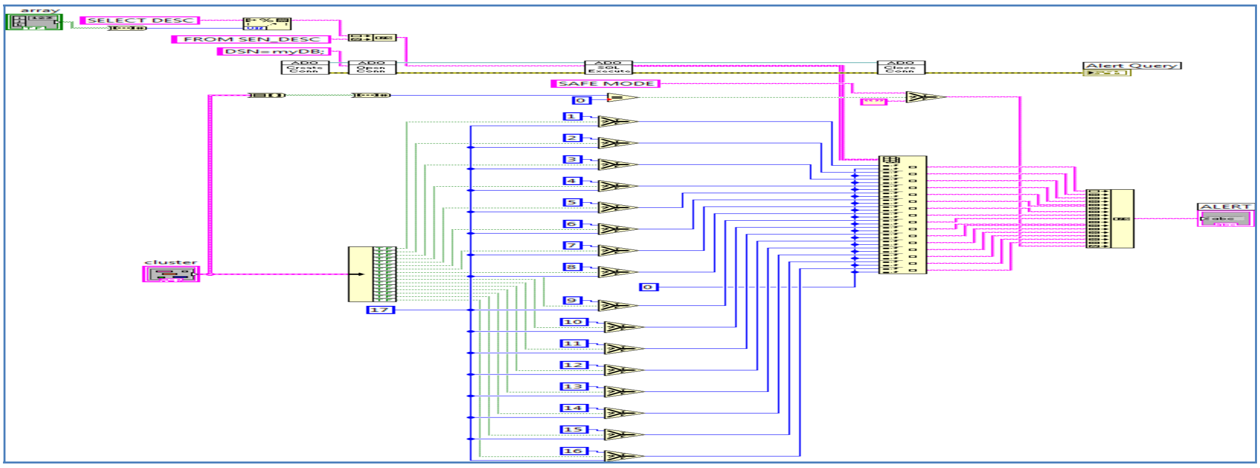


Figure (32): Alert Query

V. Conclusion

The goal of this paper is to build a complete surveillance security system which can protect any facility from any danger; also the system can be accessed through the internet. To accomplish our goal we have invented a way to transmit data between two Medias and to use database and SQL with LabVIEW.

Finally, the test of controlling VI remotely is successfully performed. It provides us a high reference value for the VI networked inspection. Since the System uses the programming technique which makes the interfaces independent and ensures the system's good transplant ability, stability, reliability, and efficiency. Our experiment shows that the system works well in both local and remote inspection LabVIEW has a TCP/IP application with "Remote Panels" So as to communicate over internet connection. In this way a local PC with LabVIEW installed can communicate over internet connection to a remote PC with LabVIEW running. The host PC can request and gain control of the LabVIEW program on the remote PC.

References

[1] Huadong Li; Yufang Zhong; Mingguang Wu, " Research on network of remote real-time surveillance system based on LabVIEW", 7th IEEE International Conference on Industrial Informatics. 23-26 June 2009, Page 60 - 65
[2] PIC Microcontrollers 1st EDITION (2008) Milan Verle, mikroElektronika
[3] PIC16F87A Data Sheet © 2001 Microchip Technology Inc.
[4] LabVIEW™ Advanced Programming Techniques SECOND EDITION R. Better, T. Mohiuddin and M. Mawrocki
[5] LabVIEW User Manual, April 2003 Edition, National Instruments
[6] Proteus Professional PCB Design and Simulation www.labcenter.co.uk/
[7] Introduction to Microsoft FrontPage© 2007 Teaching, Learning, and Technology Center Texas Tech University
[8] Microsoft Open Database Connectivity (ODBC) http://support.microsoft.com/kb/110093
[9] Introduction to Microsoft Office Access 2003 © the George Washington University Washington, DC 20037
[10] Standard Language for accessing and manipulating databases SQL http://www.w3schools.com/SQL/sql_intro.asp
[11] Internet Applications in LabVIEW: http://www.jeffreytravis.com/books/ialv.html 2006

Dr. Basil Hamed is Assistant Professor of Electrical Engineering Department, Islamic University of Gaza, Palestine, since 1999. He has Bachelor Degree in Electrical Engineering from New Mexico State University, NM. USA in the year of 1989, he received Master degree from University of New Orleans, La. USA in the year of 1992, and earned his PhD from New Mexico State University, NM USA in the year 1999. He has 15 years of teaching experience and has published many papers in national and international journals. His fields of interest include Control Systems, Fuzzy Control, Simulation & Modeling, FPGA, Signal and Image Processing.