

A Home Automation Control System Based on LabVIEW

Libo Deng

CCTEG Xi'an Research Institute, Xi'an, China

Email: 122744403@qq.com

How to cite this paper: Deng, L.B. (2024) A Home Automation Control System Based on LabVIEW. *World Journal of Engineering and Technology*, 12, 501-521. <https://doi.org/10.4236/wjet.2024.123032>

Received: March 26, 2024

Accepted: June 27, 2024

Published: June 30, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The objective of this project is to explore the possibility of using X-10 and LabVIEW to control the device in the house. Based on the serial port communication of LabVIEW, the X-10 module can be programmed by using the X-10 commands. Through the power line, all the devices connected to the socket will be controlled. Without replacing the existing wire, it must be an easy control system for the user who has no experience in electronics or communication engineering. Actually, this is a quite practical X-10 home automation system.

Keywords

Home Automation, LabVIEW, X-10 PROTOCOL, Graphic Programming, Communication

1. Introduction

The concept of “home automation” or “intelligent home” is not new. The earliest integrated home system can be traced back to the 1960s. [1] A number of standards are available all around the world in terms of regional and international standards such as X-10, Smart House, Home Bus System (HBS), Lon-Works and Consumer Electronics Bus (CEBus). (These standards will be described in the following chapter). All of these standards may not be compatible with one another because of the operating principle. Due to the huge cost, most people do not incorporate these technologies into their homes in the past. [2] Bill Gates, a famous IT man, spent more than £100 million on his smart house that controls lighting, digital art and security automatically. **Figure 1** shows the basic structure of intelligent home.

However, with the development of electronic technology, electronic devices are much cheaper than in the past and most industries invest in developing ap-

plications for smart home including remote control, Computer Interface, Lamp Module and compatible software control for existing operating computer systems. Even so, it is still not free and easy for someone who has no experience in electronics and computing.



Figure 1. Overview of home automation [3].

Home automation is truly about connecting lots of different devices to the network, including light switches, thermostats, door locks or water sensors. It is a natural progression for operators to be able to develop an integrated security and automation IP platform for the home. From the perspective of a service provider (**Figure 2**), in the next ten years, the automated home system will be used widely in both Europe and North America.

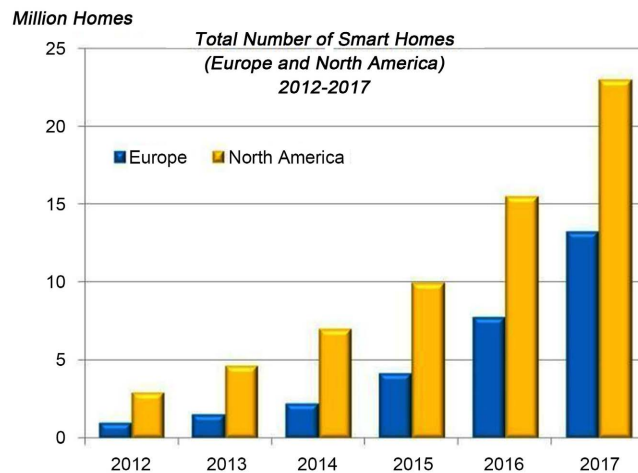


Figure 2. The number of Smart Homes [4].

Thus, it is necessary to research this project, which is based on the X-10 communication protocol to control the devices (such as lamps) by using the Virtual Instrumentation. Speaking of the Virtual Instrument, it is no doubt that

LabVIEW is a development environment for graphic programming language. In addition to providing a functional and beautiful user interface, the programming of LabVIEW is easier than traditional programming like C or assembly language (This part will be described specifically in the following chapter). The PC serial portal or USB is normally used for sending and receiving data across the mains via an interface board including setting the address and the statement of device. Basically, the final design will achieve that the lamp in different addresses can be controlled by the LabVIEW, necessary information consists of room name, lamp name, the status of lamp (“on” or “off”) and the execution time.

2. Design of Home Automation Control System by Using LabVIEW

2.1. Overview

In this project, X-10 modules will be controlled by using the Virtual Instrumentation development software named LabVIEW, which is based on graphical programming language. The software uses graphical blocks instead of text like C+ to programme the hardware and create a beautiful user interface to control the system easily. [5] The X-10 controller module will communicate with the PC via a serial port. For LabVIEW, it provides some block diagrams to support the serial communication system. According to the X-10 command, LabVIEW will send and receive the signal with the controller module and the power line will be programmed by it including the function code, house code, unit code and checksum code. Once the fundamental design has been finished, it is necessary to add a mobile control for it. People can control the device anywhere and any-time. To do this part, an Internet server will be built to run the system online. **Figure 3** shows the overview of design with block diagrams.

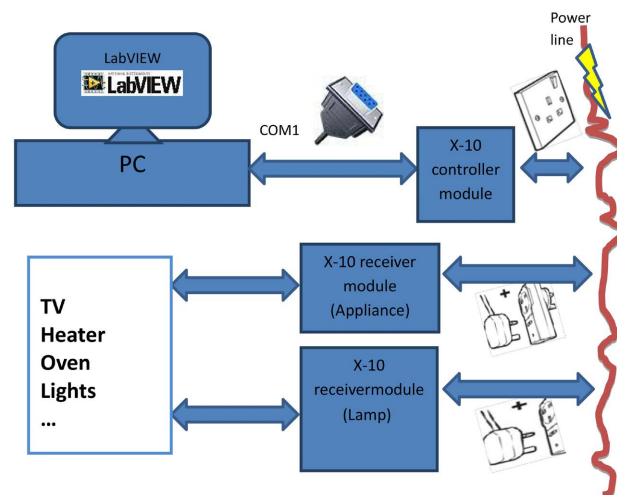


Figure 3. The block diagram of design.

2.2. Software Selection

This project will use graphic programming language, and LabVIEW is a very

advanced graphic programming software, which is using icons instead of text to program the application. Compared to the text-based programming languages, LabVIEW uses data-flow programming, which means the flow of data determines the execution. But the text-based programming is that instructions determine the program execution. For example, if people want to build a user interface with some indicators and controls, they can just drop the relevant blocks into the front panel and connect them to the block diagram interface. Block diagram interface is known as the place to program. People can add code using graphical icons of functions to control the devices that located on the front panel.

User can find some structure blocks, which have the same function as other programming languages, such as while loop, for loop, sequence structure and so on. Lots of indicators can be chosen from the indicator category like LED and string display. In a word, LabVIEW is not only a quite simple programming tool but also it can provide a nice front panel at the same time. **Figure 4** and **Figure 5** show a simple example to describe the operation of LabVIEW.

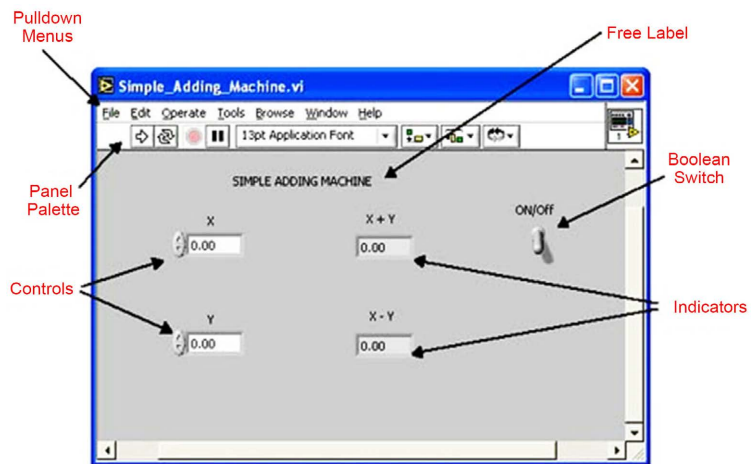


Figure 4. The front panel.

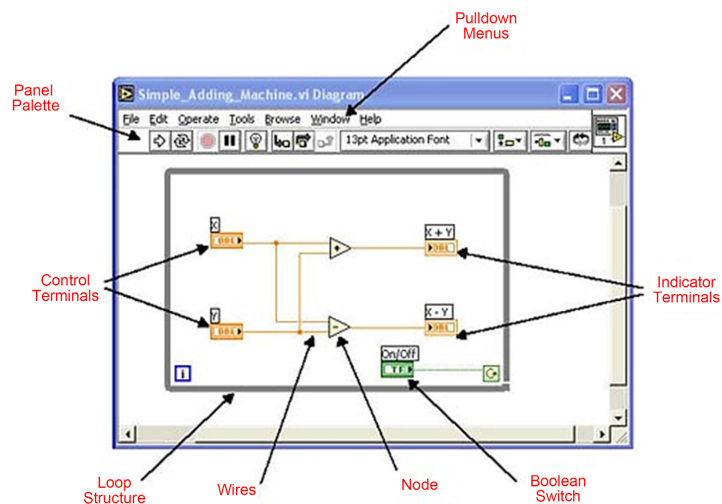


Figure 5. The block diagram.

Figure 5 is a screenshot of front panel for adding machine VI (Virtual Instruments). User just selects two controls “x” and “y” and two indicators “x + y” and “x – y”. There is a Boolean switch needed to start or stop the programming.

Figure 5 shows the block diagram that has been described in the previous paragraph. Connecting two controls and two outputs to the “add” and “minus” function icons, a while loop is for running the programming inside continuously, which is controlled by a switch. Now, running the programming and entering any number in the control blocks, you will get the result in the indicator.

2.3. Serial Port Communication on LabVIEW

There are four parameters required in serial communication, which are the baud rate of the transmission, the number of data bits per character, the sense of the optional parity bit, and the number of stop bits. In LabVIEW, serial port communication is supported by some functions and VIs, which is under the I/O » Serial sub-palette. **Figure 6** is the VISA configuring serial port VI that is used for initializing the port identified by VISA resource name to specify the settings. Resource name is the COM port in the PC. [6] Timeout is for setting the time out value of serial communication. Baud rate, data bits, parity and flow control are the specific parameters of serial communication.

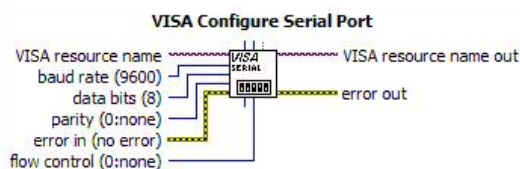


Figure 6. VISA configure serial port VI.

Figure 7 shows the basic structure of sending a string on LabVIEW by using the serial communication. The VISA configuration with COM1 and set to 9600 baud, 8 data bits, no parity and 1 stop bit. VISAWrite function sends the command to the input of VISARead function. VISARead receives the command and indicates by a read buffer string. The final step is closing the serial port.

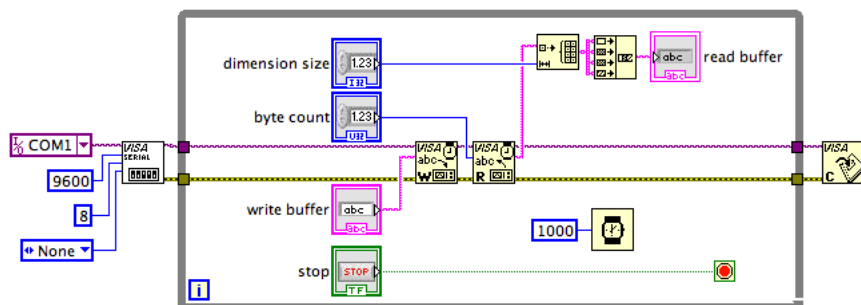


Figure 7. Send a string by using serial port.

2.4. Transmit-Receive X-10 Commands

Three parameters are used to identify the device and the function to control,

which are house code, unit code (device code) and function code. The house code is from A to P and the unit code is from 1 to 16, the function code depends on what the function user wants to configure. CM11A is a X-10 transceiver module and all the code of CM11A with respects to binary value are shown in **Figure 8**. The details of this hardware will be described in the next chapter.

<i>House code</i>	<i>Device code</i>	<i>Function</i>	<i>Binary value</i>
A	1	All Lights Off	0110
B	2	Status Off	1110
C	3	On	0010
D	4	Pre-set Dim (1)	1010
E	5	All Units On	0001
F	6	Hail Acknowledge	1001
G	7	Bright	0101
H	8	Status On	1101
I	9	Extended Code	0111
J	10	Status Request	1111
K	11	Off	0011
L	12	Pre-set Dim (2)	1011
M	13	All units off	0000
N	14	Hail Request	1000
O	15	Dim	0100
P	16	Extended Data Transfer	1100

Figure 8. X-10 commands.

The X-10 transmission format from PC to the transceiver module is shown on **Figure 9**. Firstly, the PC will write a header code to the interface, which is including the house code and unit code. Then the interface will checksum the value and writes back to the PC. If PC acknowledges the code from the checksum, the interface will be ready for receiving the data.

<i>Number of bytes</i>	<i>PC</i>	<i>Interface</i>
2 bytes	Header:Code	Checksum
1 byte	Acknowledge	
1 byte		Interface ready to receive

Figure 9. The transmission format of X-10.

The header byte (8-bit) defined as:

Header code:

7 6 5 4 3 2 1 0

<Number of Dims> 1 F/A E/S

Number of Dims means the dimming level of an X-10 module and his function is only available for a lamp module. This value is between 0 and 22 to identify the percentage of dims to be transmitted. For example, if the value transmitted is 22, it means that the percentage of dims is going to be 100%. Bit 2 is always set to "1" to make sure that the interface is able to maintain synchronization. [7] F/A distinguishes the code is for function (1) or address (0). E/S defines if the following byte is an extended transmission (1) or a standard transmission (0). So, if the address code requests to be sent, the header bytes will be 0x04. In an-

other hand, if the function code requests to be sent, the header bytes will be 0x06.

Address and function code:

7 6 5 4 3 2 1 0

<House code> <Device code> **Address code**

<House code> <Function> **Function code**

When the interface receives a data from PC, it will add all the bytes together and return a byte called checksum. If the checksum is correct, the PC should return a value 0x00 to prove the transmission should take place. When the transmission has happened, the interface will return 0x55 to the PC to show its is ready for receive transmission. The example below shows that turn on the lamp in A1. (See **Table 1**)

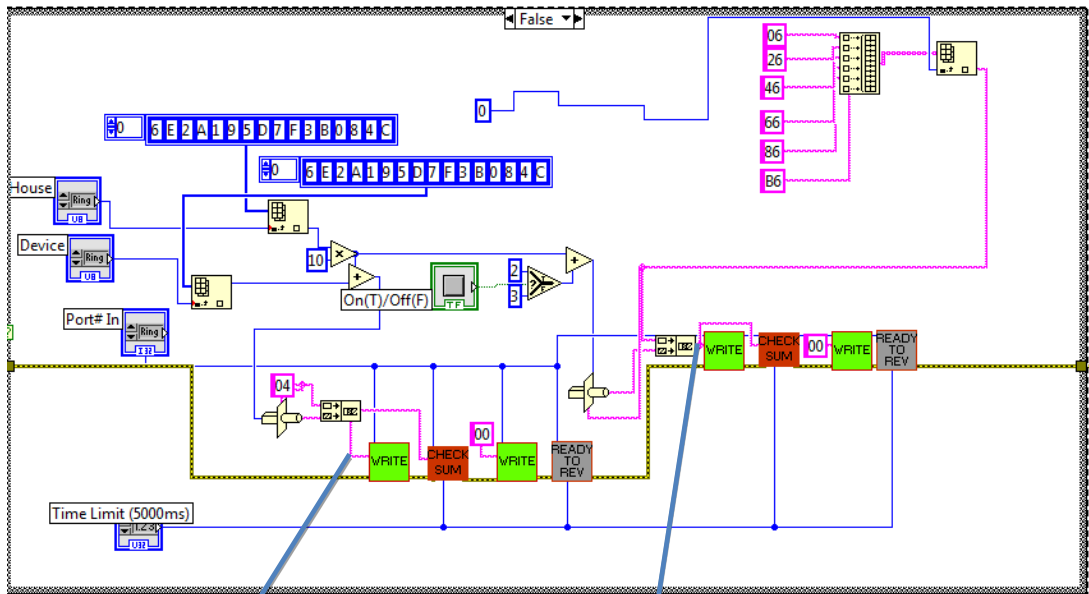
Table 1. How to turn on the A1.

PC	Interface	Description
0x04, 0x66		Address A1
	0x6A	Checksum 0x04 + 0x66
0x00		OK for transmission
	0x55	Interface is ready
0x06, 0x62		A1 ON
	0xEE	Incorrect checksum (0x06 + 0x62 ≠ 0xEE)
0x06, 0x62		Re-send the function
	0x68	Checksum
0x00		OK for transmission
	0x55	Interface is ready

2.5. Implement the X-10 Commands in LabVIEW

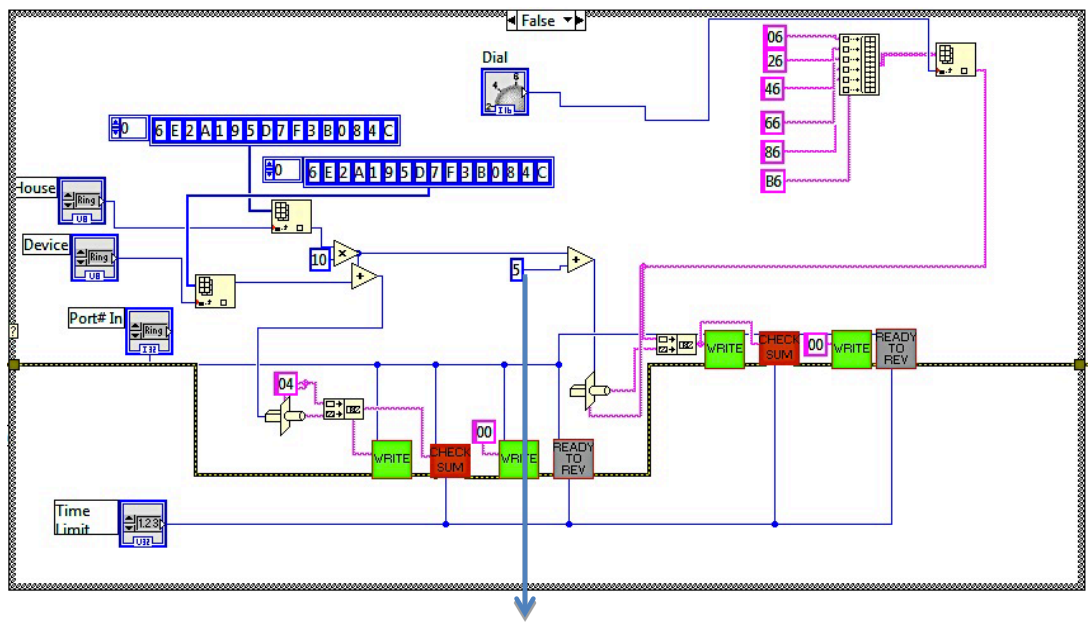
It is better to use selective ring to select which house and unit the user want to control, because it is quite difficult and complicated for users to enter the Hex value every time. So, this design here is using an array to store the entire unit and house code, when the user select the device on the ring, it will index the relevant Hex value from the array. The Boolean switch is used for control the devices ON/OFF. If the switch is turn on, the devices will turn on. If the switch is turn off, the devices will turn off. Each of processes in X-10 communication has been created as icons, which is named sub-VI. As is shown in **Figure 10**, there are four sub-VIs for this procedure including “write”, “checksum”, “read” and “ready to receive”.

Similar to **Figure 10**, the programming of DIM/BRIGHT mode is shown on **Figure 11**. The step of identifying the address is completely some as the ON/OFF mode, but the function code is different. In the X-10 protocol, the binary value for DIM is 0100, which is 0x04 in Hex. The binary value for BRIGHT is 01001, which 0x05 in Hex. Thus, changing the value that the arrow point out controls DIM/BRIGHT mode.



0x04, address code 0x06, function code

Figure 10. ON/OFF mode.



"4" for DIM and "5" for BRIGHT

Figure 11. DIM/BRIGHT mode.

2.6. Edit the User Interface (Front Panel)

Once the basic control programming has been finished, it is necessary to build a functional user interface. For example, some indicators, buttons and joystick should be used to make the control easier and nicer. In this part, the security system, devices-display system, devices-selection system reservation system and customized option system will be developed. Each part will be introduced separately.

2.6.1. Security System

The security system is aimed at protecting the home automation system from unidentified log in. In some cases, the user does not want someone else to run the system. So a password is needed for the users to log in the system only. **Figure 12** shows the design of this part.



Figure 12. The front panel of security system.

The control panel will not appear until the password is correct. The user enters the password first and presses OK. If the LED turns to green, it means log in successfully. Otherwise it will stay at red colour and wait for user types the correct password. There is a possibility that householders forget their password. In consideration of this case, a password-reset option will be created.

Since the LabVIEW can read the external document like XML or text format, the system provides a changing password file, which is the text format. The length of password has been set by 6-character. If the user wants to change the password, just open this file and type new 6-character. Without doubt, this file should be kept very safety. (See **Figure 13**)

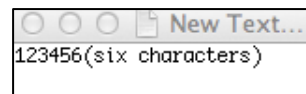
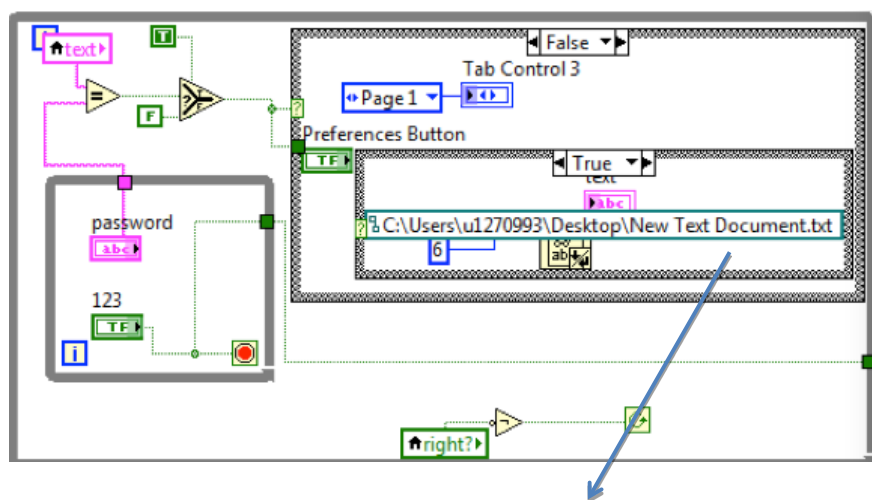


Figure 13. The password reset file.

Figure 14 shows the block diagram of this part. By using the “equal?” function block, this part is not complicated as people imagine. It just judges two strings are completely same or not. If they are same, the “equal?” block will output “1”, which can make the programming continuous without stop.



Where the password-changing file is

Figure 14. The block diagram of security system.

2.6.2. Device-Display System

In a house, varieties of devices with different types need to be controlled. For example, heater, microwave oven, washing machine, light, air-condition, TV and play-station. The user interface should tell users which device is controlling and what the status it is running. In this design, it modifies the indicator for heater, oven, washing machine and light. Basically, a bulb picture will stand for the statement of light. If the user wants to customize the devices, there are two options for them, bulbs for lights and appliance module for appliance. Different devices will be provided for user, which is shown on **Figure 15**.



Figure 15. Some devices indicator (default).

Adding these pictures into the picture ring, it will indicate the devices that user selected. Dropping the picture ring into the front panel and changing it to read mode. Then copy the picture from PC and import the picture to the ring. After this, the picture ring will give an order for each picture. Using the number controls the pictures with different choice. **Figure 16** shows where the picture ring is.

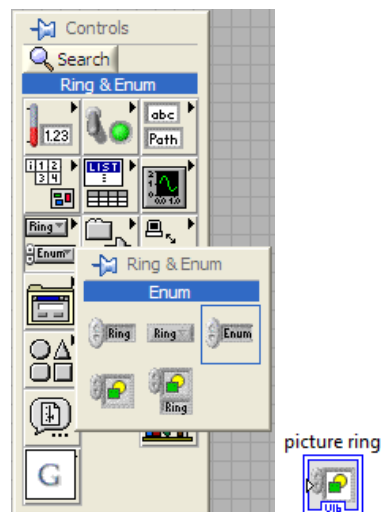
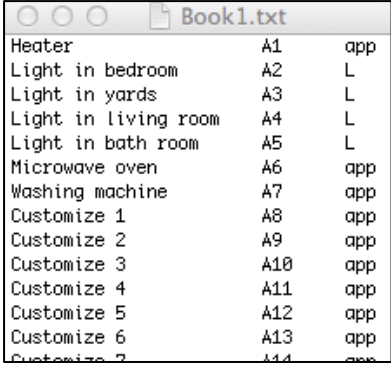


Figure 16. How to find the picture ring.

2.6.3. Devices-Selection System

For convenience, this system provides a devices-selection system for users to choose which device they want to control. To do this, there is a document needed which is for user to edit the devices, here is used by another text file. There are 3 columns in this file, which are device name, address and type. Note that the appliances will representative as app and the lamps will representative as L. For example, the device name is heater, the address is A1 and the type of device is appliance. (See the **Figure 17**)



Device Name	Address	Type
Heater	A1	app
Light in bedroom	A2	L
Light in yards	A3	L
Light in living room	A4	L
Light in bath room	A5	L
Microwave oven	A6	app
Washing machine	A7	app
Customize 1	A8	app
Customize 2	A9	app
Customize 3	A10	app
Customize 4	A11	app
Customize 5	A12	app
Customize 6	A13	app
Customize 7	A14	app

Figure 17. The customised file for user.

Once the edit file has been finished, these settings will be bound in a selective ring. If the user clicks the drop-down list, all the devices will appear on the screen. There are 7 devices has been set up already as the default. Heater, Light in bedroom, Light in yards, Light in living room, Light in bath room, Microwave oven and Washing machine. (See **Figure 18**) In addition, there are lots of customize options for user to add more devices in the room. Just a four steps to set up:

- 1) Open the device edition file
- 2) Change the name of device
- 3) Modify a new address for it (house code: A-P, unit code: 1-16)
- 4) Choose the type of this device (appliance: app, lamp: L)

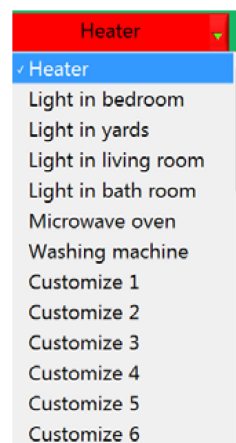


Figure 18. The device selection system in the front panel.

From **Figures 19(a)-(c)**, it shows how the display system and device selection system work.

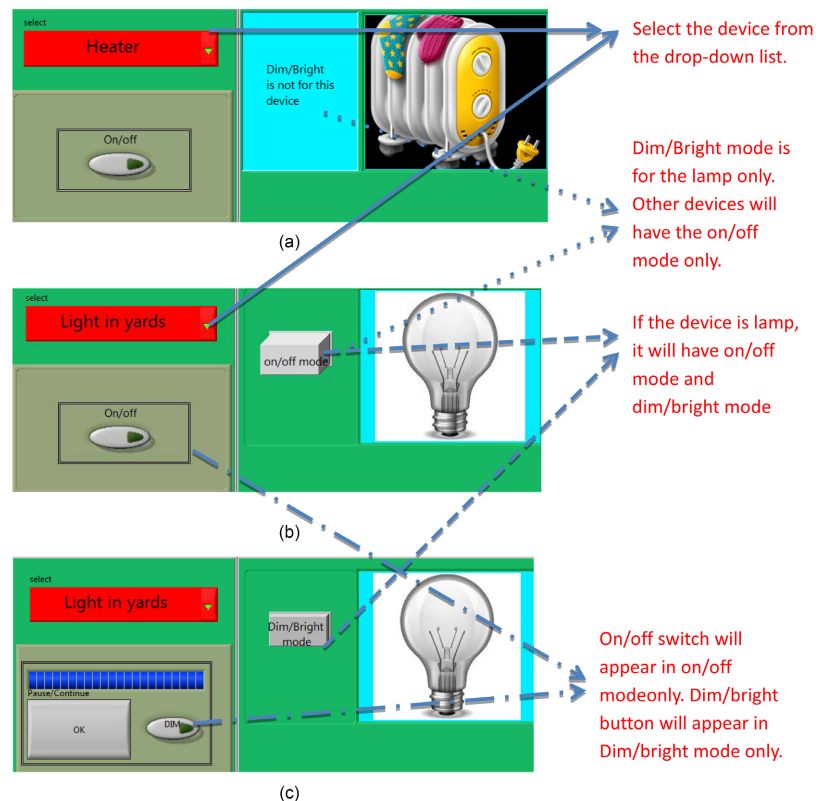


Figure 19. (a) The front panel for appliances; (b) The front panel for lamps; (c) The Dim/Bright mode for lamps.

2.6.4. Reservation System

Reservation system actually is a time control option. The operation of this part is when the user needs to make a reservation on the working of devices, just turn on the time select switch. Then timer interface will appear on the system and wait for user to set the ON time and OFF time. If people want the system keep the time setting for some time, there are 8 buttons, which provide options for users to determine when they want the system running as the setting before. The 8 buttons are Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, Every day. The interface of this part is shown on **Figure 20**. The time setting is quite simple with a few steps. Firstly, click the time configuration button that looks like a calendar icon. Then change the time including month, year, hours and minutes. Finally, click the OK button.

In the block diagram, as shown in **Figure 21**, it shows how to get the current time from LabVIEW. The current time can be achieved by using the block, which is named “get time/date”. Then connect this icon to the in range and coerce function. The upper time limit and the lower time limit will be set in this function to judge the time is in range or not. If the time is in the range, the light will be turn on. Otherwise, the light will be turn off.

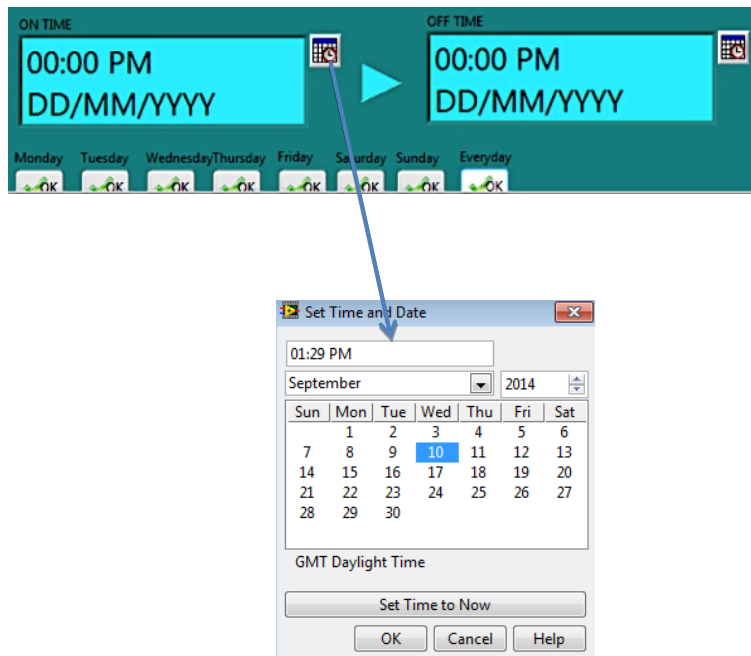


Figure 20. The time select panel.

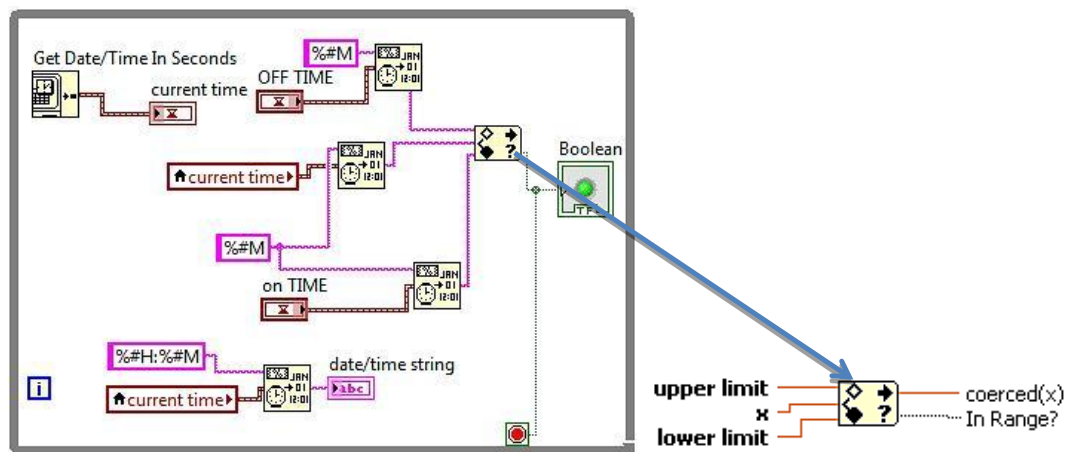


Figure 21. The block diagram of time selection.

If you change one of the data types to an array, you must change the remaining data types to arrays to avoid broken wires. However, you can wire an array to x and two scalars to upper limit and lower limit. You also can wire double precision, floating-point and integer data to the input. The output will be replaced by a Boolean value. If the input data is in range, the Boolean value is true. If not, the Boolean value is false.

2.6.5. The Final User Interface

The final user interface is shown as Figure 22. There is a house picture in the front waiting for users enter the password to log in the system. Once the password is accepted, this picture will disappear and the control panel will come out showing all the options to the user. To become a product, a specify instruction

should be provided by the designer. So, an instruction is produced for people who start to use the system in first time. There two elements there, the advantage of this system and the guide how to use it. In the next section, this user interface will be created as an “.exe” application, which can work in the PC without the LabVIEW. And an installer package will be generated for the user to install this application anywhere.

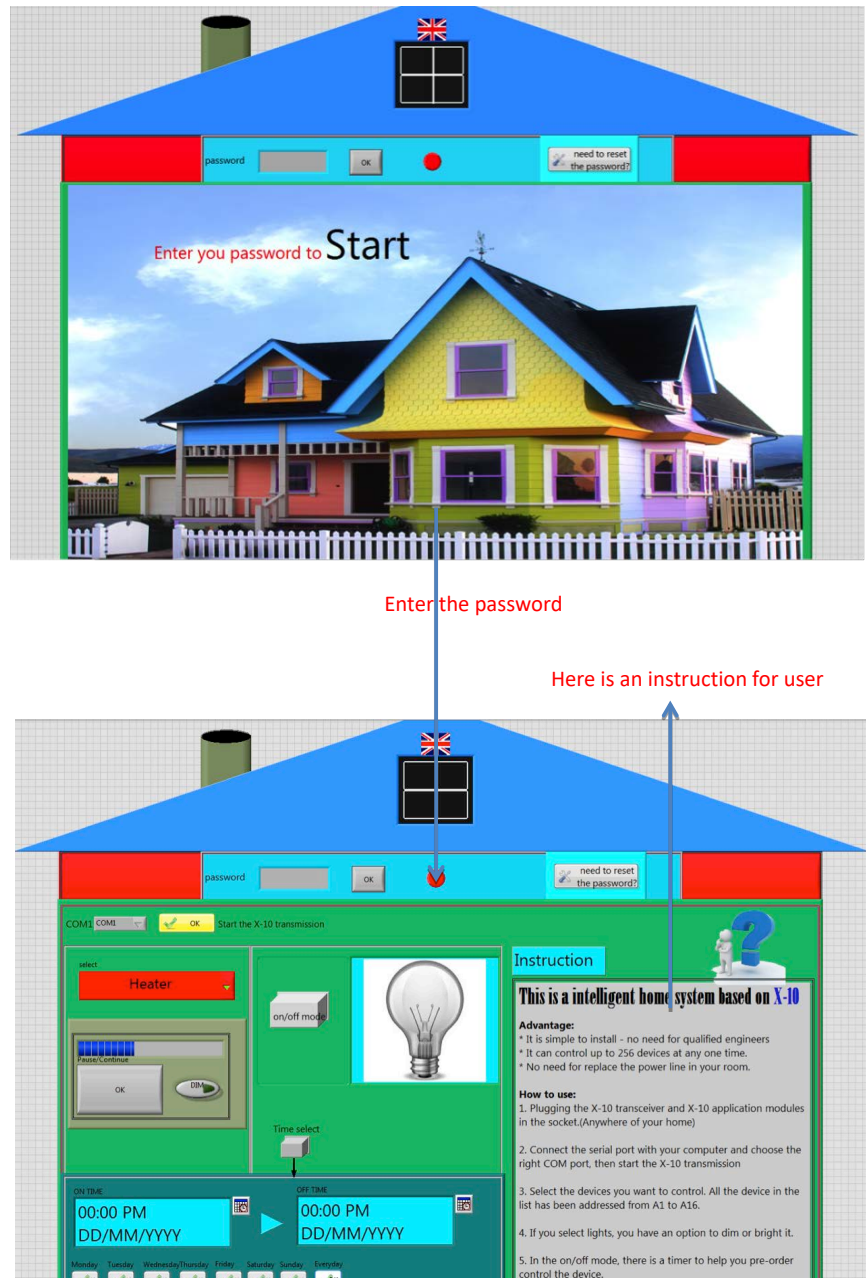


Figure 22. The final front panel.

2.7. Create the .exe Application and Installer for the System

The executable application is independent VIs and file build together into one

program, which can be run in the computer. The installer package includes the executable files, the device driver and the LabVIEW run-time engine environment. This package can be installed on any other PC, just like the way to install normal software on windows. **Figure 23** shows the executable file and the general installer package.

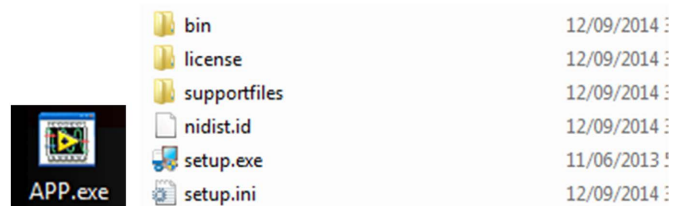


Figure 23. The executable application and the installer package.

2.7.1. The “.exe” Application

LabVIEW offers an option for engineers to build an “.exe” application for their project, because the user may not have the LabVIEW in their own PC. In most cases, the “.exe” application is protecting the system. Due to the “.exe” application will not allow user to see the programming (block diagram), user can only run the system without changing the design from the programming window. If the user access the block diagram unintentionally and change some program they do not understand, this may crash the program and the system will be destroyed. So, the “.exe” application is very necessary to build. **Figure 24** shows the steps to build the .exe application. It looks like simple, but sometime not. If the user was interested in programming, they might access the *LabVIEW Help* by selecting **Help»LabVIEW Help** in LabVIEW to solve the possible problems.

2.7.2. Installer Package

Right click the *build specification* in the project and select the installer. In the installer property, configure the product name and the source files. Create a folder to save the installer. The most important step is that adding the additional installer to the package. The reason is that people may use different version of operation system, and some drivers for LabVIEW will be included for special use. For example, in this project, the serial communication driver and the VISA (Virtual Instrument Software Architecture) drivers must be configured in the package. After building the installer package, you will find a folder in the destination, which called installer. If you open that file, you will find that there is an application inside named *setup.exe*. When the user want to install the application to another computer, just click this file and install it step by step. (**Figure 25**)

2.8. Create a Network Server for the System and Generate a Web URL

Use the LabVIEW Web Server to publish images of front panels from a VI or a stand-alone application to the Web. In the default mode, after the web-server has been enabled, all VIs and applications are visible to all Web browsers.

However, you can control browser access to the published front panels and configure which VI is visible on the Web. Not that to display front panels on the Web, the VI must be stored in the memory of the computer.

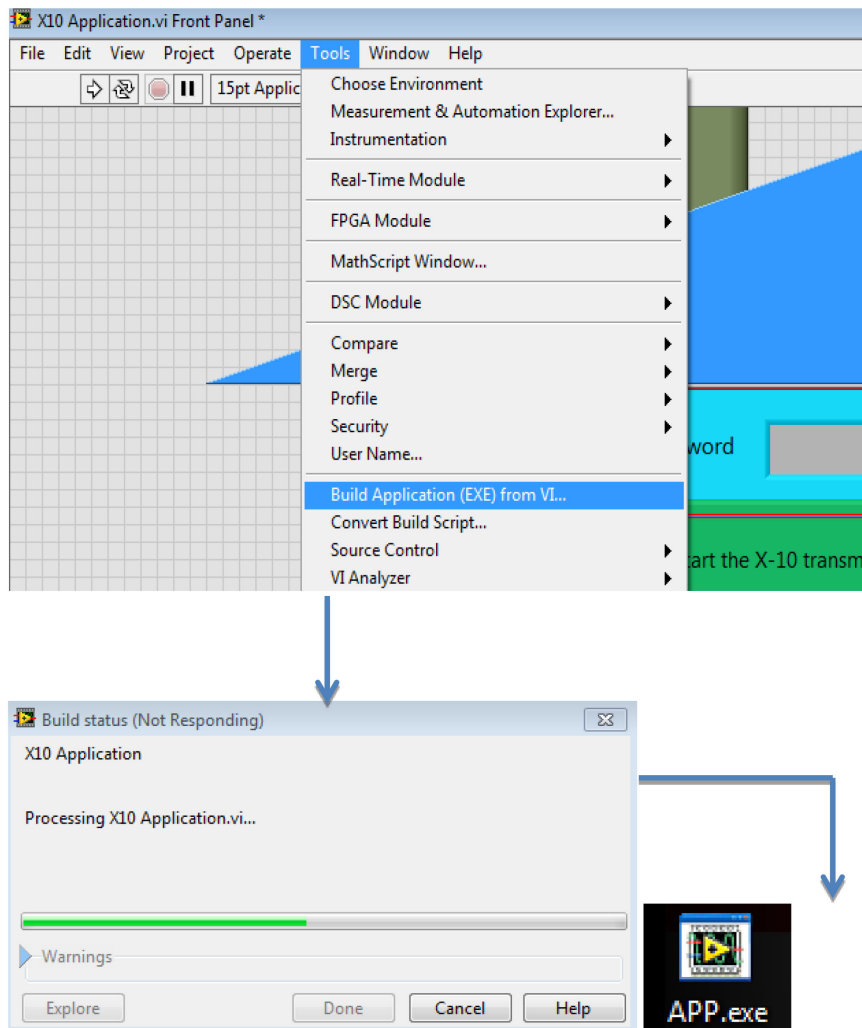


Figure 24. The basic step to build an executable application.

The JPEG and PNG image formats can be generated for the front panels by the web sever. The JPEG format compresses graphics well but can lose some graphic detail. This format works best for photos. Use the Web Publishing Tool to create an HTML document and embed static or animated images of the front panel or to embed a front panel in an HTML document so a client computer can view and control the front panel remotely. Use the **Snapshot** option in the Web Publishing Tool to return a static image of the front panel of a VI currently in memory on the server computer. Use the **Monitor** option in the Web Publishing Tool to return an animated image of the front panel of a VI or application currently in memory on the server computer. Use the **Embedded** option in the Web Publishing Tool to embed a front panel in an HTML document so a client computer can view and control the front panel remotely. [8] (See **Figure 26**)

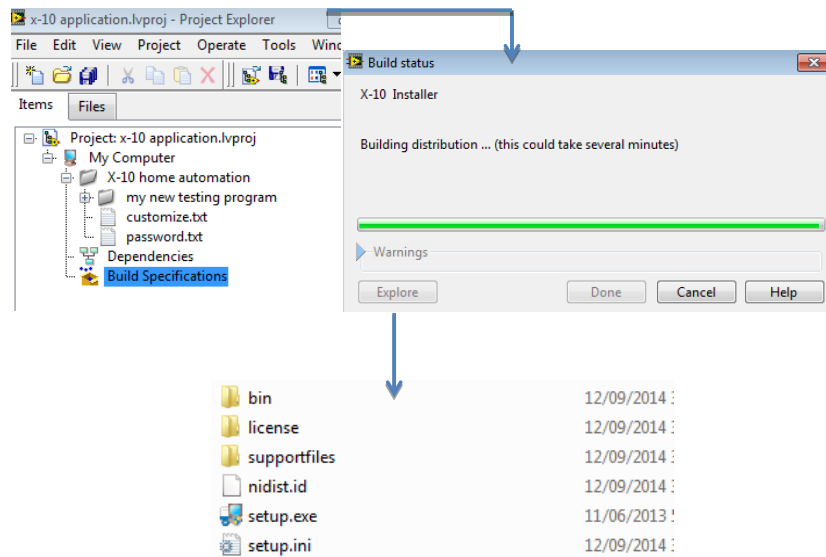


Figure 25. The basic step to generate an installer package.

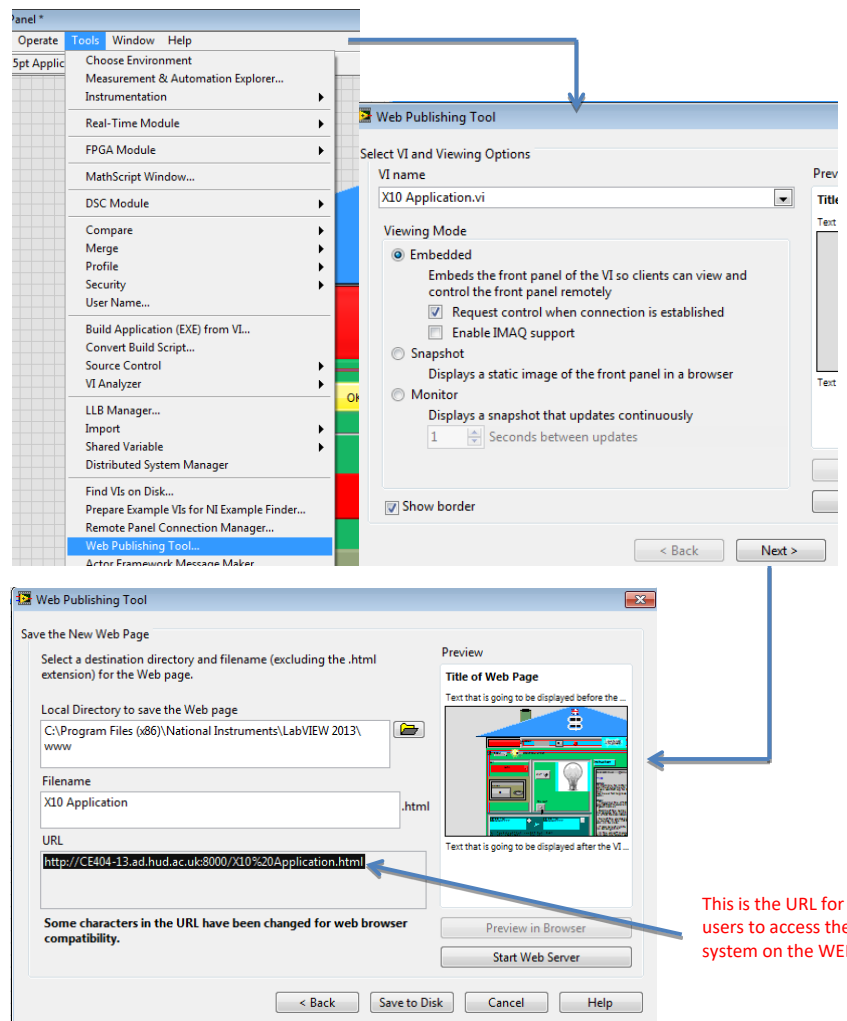


Figure 26. Create a URL for the design on LabVIEW.

2.9. Hardware Description

In this project, the hardware is not going to design. All the modules are compatible with X-10 will be bought, in terms of the X-10 controller (CM12U), Lamp module (LM565) and Appliance module (AM7466). The X-10 controller connects to the PC via a serial port, and Lamp and Appliance module plug in the room socket. This chapter will describe the specific of these modules and how it works.

2.9.1. CM12U X10 Computer Interface

With the CM12U X-10 Computer Interface, users can program and control their lights and appliances at home. The manufacturer has created software for it, when people buy this device, it will be including in the box. This project will use the X-10 commands to program this controller instead of using this software. Due to this device provides self-program options for user, a single X-10 command can be used to trigger and program the power line. **Figure 27** shows the picture of CM12U computer interface.



Figure 27. CM 12U X-10 controller.

Feature:

- Available as either 9 Pin Serial or USB connection
- Control X10 devices directly from your PC
- Allow user to self-program
- Compatible with all the X-10 commands

The CM12U unit plugs into the serial port of any PC running Windows. The socket on the front of the unit is pass-through power outlet. The X-10 signal will send and receive through the serial port, and the power line will be programmed through the socket. 2xAAA batteries are required to keep the RAM intact. The mains electricity is not the power supplier for the RAM. If the PC is disconnected, then batteries are required to make the RAM working.

2.9.2. Appliance Module

Appliance module is used to control the devices that have On/Off mode. It is a receiver unit that works as a remote controlled relay. You can plug it into any socket in the house and control the unit attached to it through X-10 commands.

While lights have a special mode, which is the Dim/Bright mode. If the light plugs in this module, the Dim/Bright mode will not be working. In addition, the appliance module can switch incandescent lamps up to 500w.

The switch on the load device can still control itself locally. However, do not forget to leave the local switch to “On” position if you want to receive the X-10 commands. Otherwise the X-10 controller cannot control the device. **Figure 28** shows that appliance module was used in this project, which is AM7466. On this module, the load devices will be addressed by the house code (A-P) and unit code (1-16).

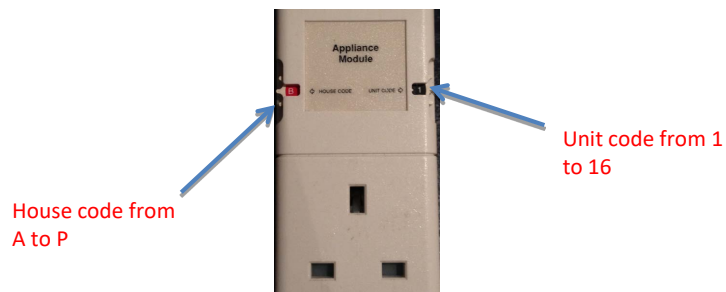


Figure 28. AM 7466 appliance module.

2.9.3. Lamp Module

X-10 Lamp module is used by lights only, because it has a special function, which is the dimming control of incandescent lighting. This is the LM565 lamp module that operates with 230V AC and 40W to 300W incandescent lamps. It is not suitable for appliances or energy saving bulbs.

As this module is simply inserted into wall sockets like an ordinary plug, there is no installation to be carried out. Just adjust the wheels on the front to desired device address and leave the load devices switches in the “on” position, which is as same as appliance module mentioned on last section. **Figure 29** shows the LM 565 lamp module.

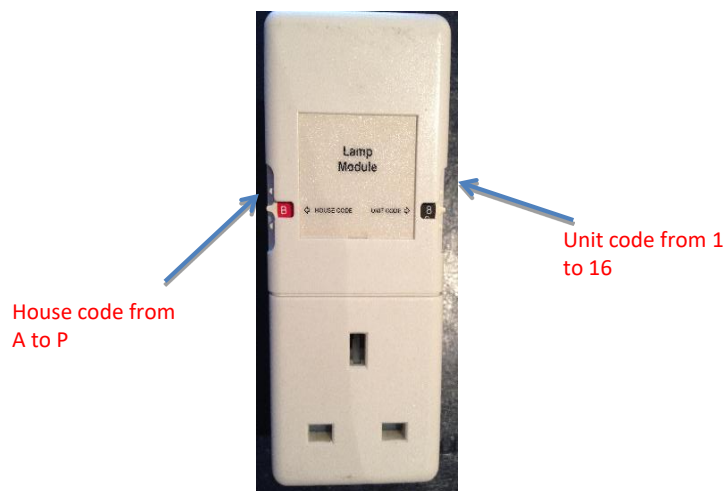


Figure 29. The Lamp Module LM 565.

2.9.4. The Connection of Hardware

Figure 30 shows the overall connection of hardware. Lamp module and appliance module are plugged in the socket communicate with the control module via the power line. The X-10 control module has a serial port that connects to the COM port of PCs. Based on the LabVIEW user interface of this project, it works successfully.

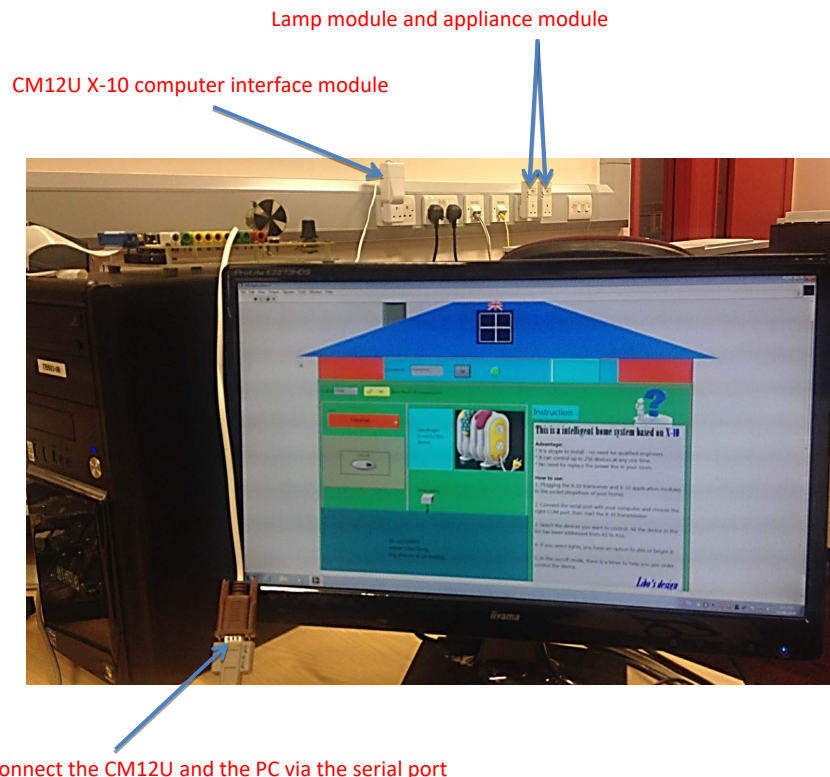


Figure 30. The total connection of project.

3. Conclusions

The software of LabVIEW was selected to program the X-10 modules by serial communication. This software built a nice user interface for people and that was the main reason why this software was chosen. To modify the front panel (User Interface), the security system, devices-display system, devices-selection system, reservation system and customized option system are all built in LabVIEW.

After the main design had been finished, an executable application was necessary for the user to make the system portable. Due to the user changing their PC or some special cases, an installer package had been generated and added to the project. To improve the practical applicability, this system was developed by the Web. It means that the user can control the system online by typing the URL on the Internet browser address bar.

In a word, the home automation system here is not the most advanced design in terms of software and hardware. However, it is very simple for the user to install and make their home intelligent. With the development of intelligent home

technology, the author believes that more and more families will install the home automation system in their houses.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Wacks, K.P. (1992) International Development of Home Automation Standards. *IEEE 1992 International Conference on Consumer Electronics Digest of Technical Papers*, Rosemont, IL, 2-4 June 1992, 274-275. <https://doi.org/10.1109/ICCE.1992.697223>
- [2] Nunes, R. and Delgado, J. (1998) An Architecture for a Home Automation System. *Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems*, **1**, 259-262.
- [3] Zahariadis, T.B. (2003) Home Network Technologies and Standards. Artech House Inc.
- [4] Douligieris, C. (1993) Intelligent Home Systems. *IEEE Communications Magazine*, **31**, 52-61. <https://doi.org/10.1109/35.237984>
- [5] X-10 Ltd. The History of X10. <https://hornerautomation.com/product/x10/>
- [6] Finley, B. (2006) Powerline Play. *Electrical Construction & Maintenance*. <http://ecmweb.com/contractor/powerline-play>
- [7] Dr. Wacks (2000) Home Automation and Utility Customer Services. Cutter Information Corporation.
- [8] Jeffrey, T. (2007) LabVIEW for Everyone: Graphical Programming Made Easy and Fun. 3rd Edition, Prentice Hall Professional.