

Design of Simulated Dot Matrix Moving Character Display Based on LabVIEW

Li Su^{1, a}

¹School of electronic engineering, Xi'an Aeronautical University, Shaanxi Xi'an 710077, China

^a717246289@qq.com

Abstract

The basic requirement of dot matrix character moving display design is to input any character in the character input control, which can generate movable fonts in the LED dot matrix display on the front panel. First, the font extraction is conducted, and the characters are converted into a two-dimensional array composed by 0 and 1, array sorting is carried out for extracted fonts in program panel to generate a moving effect, and finally the two-dimensional array quantity is converted into a Boolean quantity and sent to LED dot matrix display of front panel.

Keywords

Virtual instrument technology; LabVIEW; moving character; dot matrix.

1. INTRODUCTION

LabVIEW this programming software is different from previous programming software, which does not have a lengthy program; instead, graphical and numerical table program node and flow chart requires programmers, use the terms and concepts people understand as much as possible, "build" the required instruments. Intuitively, the user or other programmer can see it fully at a glance. LabVIEW is used to study the principle; the development efficiency when designing can be greatly improved.

The design of simulated dot matrix moving character display is to input characters in an input control, output the font of the input text in the LED dot matrix display, and generate a moving effect. First, we need to input the characters, output the outline of its characters as two-dimensional array data, and then sort this array to form a moving effect; finally, we must convert the array number into a Boolean form and deliver it to the LED display of front panel. According to the above description, it is decomposed into three parts, font extraction module, array sorting module and a Boolean conversion module; the structural block diagram of the system is shown in Fig.1.



Fig 1. Block diagram of system structure

2. FONT EXTRACTION MODULE

The function of the character extraction module is to randomly input a string of characters (Chinese characters or English) in an input character control, after the system run, a character composed of a circular indicator light one by one can be formed in a character.

Character movement only needs to shift its column, so only need to carry out one index for the input sequence, first set a For loop, in the rectangle setting of the previous module, set its "row" to 20, so N input constant to 20. The one-dimensional array cyclic shift inside For and output the previous module, and as the input of this module, open the index of the For loop, connect it, set the n terminal of the one-dimensional array cyclic shift to 1, then the new array is indexed through the For loop and output, create a 20×70 array and run the program, the result is that the input character is only moved one unit to the right, because the For function indexes each row of sequence, and the internal program moves the sequence one unit at a time, it shows that a loop needs to be added outside of the For loop, and the last output array can be fed back to the starting point and loop.

In order to complete the continuous movement of characters, first, add a While loop outside the For loop, add a shift register, connect the array input to the left terminal, and create an output control (marked as "1 output") to the right terminal, the inside is the same as above For loop connection, the output is connected to the right terminal of the shift register, and the While loop conditional terminal stop access the false constant, when running program, it finds that there is no data output, but when a display control is inserted in the output For loop and the While loop are in the output (marked as "2 output"), it can be found that the data moves quickly to the right, this is because LabVIEW is a data stream, and the connection indicates the flow of data, the data in the While loop is read once and displayed once, the data outside the While loop can only be displayed until the While loop is stopped, therefore, as long as the While loop condition terminal still meets the operating conditions, the outside of the loop will not be read. If people want to place and loop outside, it is necessary to create a property node or a local variable to pass. But logically, adding a While loop to the outside can see the character movement in the output control, so that even other programmers can see the program at a glance. The program is shown in Fig.2.

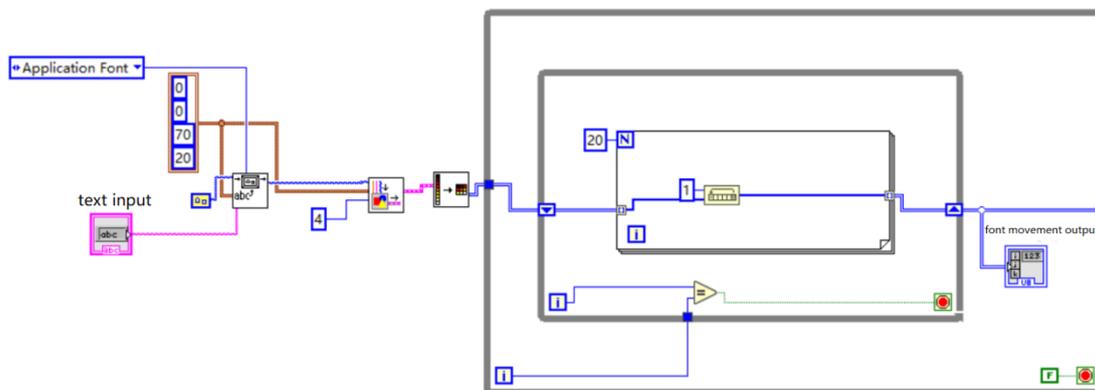


Fig 2. Font movement module mode and font module extraction module cascade

Because the data moves too fast, it is necessary to decelerate the program, add a wait formula node inside the While loop, enter constant 300 (and conduct a While loop every 0.3 seconds), and then run the program, the value in the display control slowly moves to the right. However, if the program can be run, the speed can be controlled manually on the front panel, which can enrich the application of the program.

Add a knob on the front panel, change the label to "speed", right click on the ruler in the property bar, change the maximum value to 300, and change the minimum value to 0, then turn to the rear panel and change the value input of the waiting node to just created input control, the program run, at 0, the value moves very fast, slowly increase the value, the value of the array moves slower, and complete the control of the speed.

But can the direction of character movement change? The answer is yes. Reset the n-terminal of the one-dimensional array shift to "-1", it means inserting the first value of the one-dimensional array into the end, shift the whole one unit to the left, the test run program, the whole array moves to the left and continuously.

In order to increase the overall flexibility of the system, reset the n-terminal of the one-dimensional array shift, add a horizontal rocker switch on the front panel, and change the label to "left and right" and add a comparison logic in the For loop of the rear panel, set a constant "1" at the t terminal, set "-1" at the f terminal, move the Boolean switch to the For loop and connect the conditional end of the comparison logic, and output connect the to n. Run the program to turn the rocker switch to the left, the array value moves to the left as a whole, the switch is turned to the right, the array value is moved to the left as a whole, and the array sorting module program is shown in Fig.3.

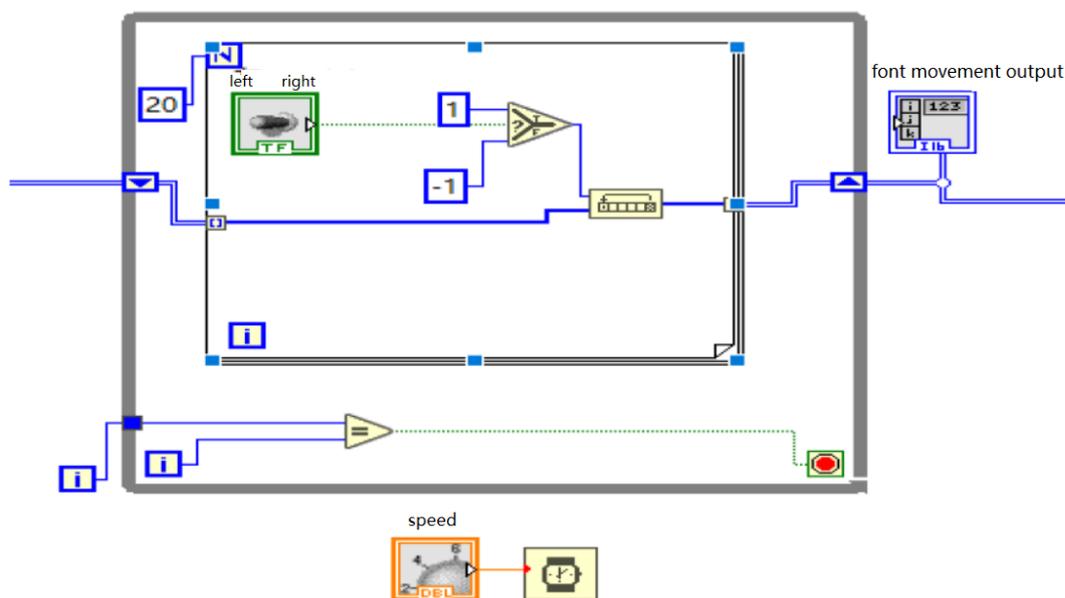


Fig 3. Sorting module program of array

So far, the array sorting module is completed.

The output of the previous module is a two-dimensional array, as the input of this module, because the "1" in the array is changed to the "true" of the Boolean quantity, and the "0" becomes the "false" of the Boolean quantity, Therefore, it is necessary to reduce the dimension of the input, first, set two For loops, in nested mode, the same word modules extract module, the external For loop conduct index for row, so set the value 20 to N, the internal For conduct index for column, so set N to 70. Since the final purpose is to convert the input 0, 1 into a Boolean quantity, it is displayed in the form of an indicator light, Therefore, a formula node of "not equal to 0?" is set in the internal For loop, through the index of twice For loop, two-dimensional array is split into value one by one, and then Boolean quantity is converted, and finally open the index of the two For loops, also open the index, pull the output, recombine the values into an array, connect the 20×70 LED dot matrix which has been set, Boolean quantity conversion module program is shown in Fig.4.

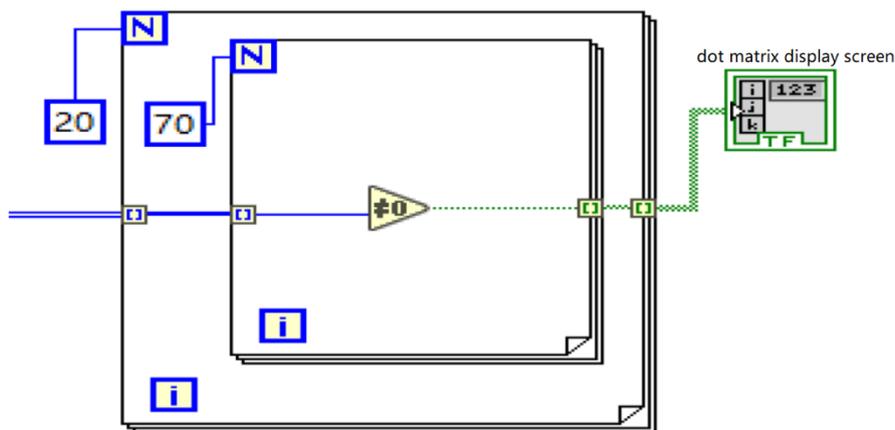


Fig 4. Conversion module program of Boolean quantity

So far, the programming of the last Boolean quantity conversion module has now been completed.

After the program runs, it is found that there is no character display, because outside the While loop, the While loop will output data after meet the stop condition, so we need to include the Boolean quantity conversion module in a loop, but the situation is different from the array sorting module at this time, if only the Boolean quantity conversion module is nested outside the While loop, no data is displayed, it is necessary to consider the data transmission problem of the module before and after, the array sorting module will transfer the transformed array data to the next module, and the next module will receive the data, through the number output by the While loop, if the loop does not stop, there will be no data output. Thus it can be seen that the array sorting module and the Boolean quantity conversion module must be included together through a large loop, and the array sorting module and the Boolean conversion module simultaneously work and stop, then the character dot matrix can be displayed.

First, we set up a large While loop to include the array sorting module with the Boolean quantity conversion module, and the LED dot matrix display is also set inside the loop, and connect with the output of the Boolean quantity conversion module. Set a false constant for the conditional terminal. It is necessary to consider the internal and external loop connection problems, if there is no such detail correction, it is impossible to output any value, because complete operation of the internal program of the block diagram when looping, after i add one, input the new value for the next cycle, if there is no internal or external contact, the internal loop once, although waiting for the external transmitting a new value, the external loop command cannot be obtained, and the next loop cannot be performed, as a result, there is only one data input and no data output.

The synchronization method between the global loop and the local loop is to add a node "equal?" in the outer While loop of the array sorting module, the counting terminal of this While loop and the external large While loop is connected "equal? Two input", the output is connected to the conditional terminal of this While loop (the "false constant" set before deletion). The principle of this setting is that when our font extraction module is transmitted into a large While loop, the loop starts, the counting terminal is incremented by 1, and the data enters the array sorting module, at this time, the loop starts once, and the first sorted array is output, the counting terminal is incremented by 1, the value of internal and external condition terminal are equal, stop, pass the shift register previously set on the While loop of the array sorting module, the current data is transmitted to the left terminal, and the output array is passed to the Boolean quantity conversion module to complete an external While loop, because this loop is set to stop

when if true (meaning it is always running), at this time, the counting terminal is incremented by 1, the conditional terminal in the While loop of the array sorting module does not meet the stop condition, then continue to run until the end of the loop, delivery and feedback, the counting add one, which meet the stop condition, running is completed, external re-run, etc., by analogy, our first overall system correction is completed.

The font shift on the LED dot matrix display must input the input character in the character string before running the program, then display on the dot matrix screen, and modify the text in the string character input control in the case of program running, there is no change in the display of dot matrix screen, so, it is necessary to make a second correction of the system, so that when the program is running, input a string of characters, which can be immediately displayed in the dot matrix screen, and can be repeatedly written to make the running program more flexible.

Through the overall analysis of the system, as well as continuous practice, it is concluded that if there is no loop, any input can only be captured once, only in the case of external loop, internal input, execution end, capture input, continue to execute, so the first module needs to be included in a while loop, but the latter two modules are included together through a large While loop, a small modification is conducted in the system, and the font extraction module is incorporated into the loop, at this time, the running program found that the second correction was completed as expected. The general program of dot matrix character movement design is shown in Fig.5.

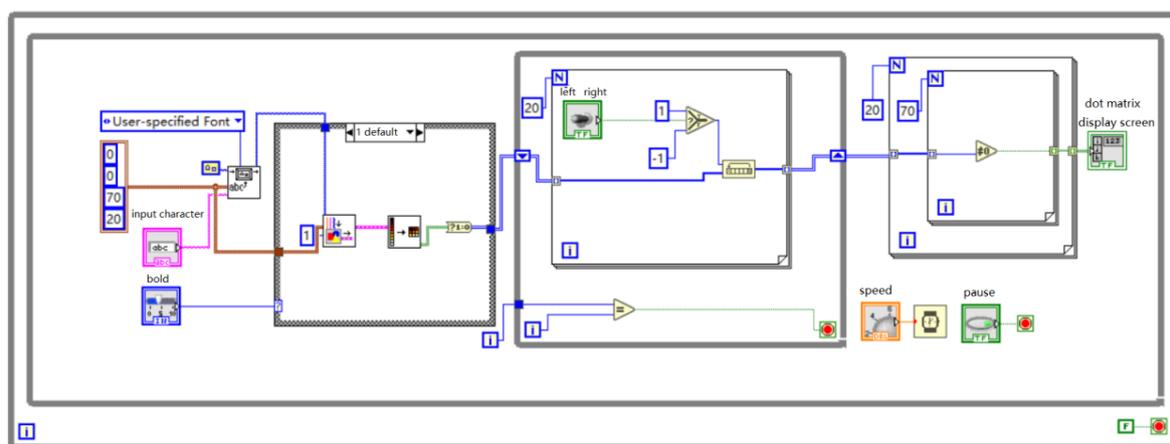


Fig 5. Total program of dot matrix character moving design

REFERENCES

- [1] Zhu Haitao. LED Dot Matrix Display System [J]. Practical Electronics, 2016 (6): 40-41.
- [2] Yang Fuqiang. Led Lattice Dynamic Display System Based on Single Chip [J]. Industrial Control Computer, 2018, 26 (1): 101-102.
- [3] Zhang Mingbo. Design of Matrix LED Display System Based on MCU [J]. Microcomputer Information, 2017, 23(5): 84-86.
- [4] Li Weiwei. Research and Design of Multi-function Control System of LED Display Based on ARM [D]. Shanghai: Donghua University, 2018:20-31.
- [5] Lei Zhenshan. Practical Technology Course of LabVIEW 7 Express [M]. Beijing: China Railway Publishing House. 2016: 90-104
- [6] Li Guangming, Li Gu. Construction of Networked Virtual Instrument System Based on LabVIEW [J]. Control and Instruments in Chemical Industry. 2017, 18(1):2-6.