

Arduino-based desktop robot control system design

Wenbei Liu ^a, Huibin Fu ^b and Xiangmei Du ^c

School of Mechanical and Electronic Engineering, Shandong University of Science and Technology, Qingdao 266000, China.

^a 1273179928@qq.com, ^b imasmallfish@163.com, ^c 879864874@qq.com

Abstract

Arduino is an platform that based on the open source code, the platform consists of hardware and software components. Firstly, write the arduino program and then design the labview front panel and block diagram, focus on the serial communication between arduino and labview. Finally, complete the arduino-based desktop robot control system using labview as the upper computer software.

Keywords

Arduino, labview, serial communication, desktop robot.

1. Introduction

Arduino is an open source electronic prototype platform that is flexible and easy to use [1]. Labview is a program development environment. The obvious difference between labview and other computer languages is that other computer languages use a text-based language to generate code, while labview uses a graphical editing language to write programs. The resulting program is in the form of a block diagram [2,3]. With the development of modern industry, more automation is needed in industrial production [4]. Most of the existing robotic arm control systems are complex and expensive, and it is difficult to meet the individual needs of the small factories [5]. However, small factories are an important force for economic development. How to quickly develop a simple and practical robot control software for upper machine has become an urgent problem to be solved.

2. Arduino programming

The Arduino program design mainly includes the definition of the servo and its interface, the serial communication with labview, data processing and so on, focusing on the serial communication between arduino and labview, making it possible for the arduino to read commands issued by the computer and feed back the results to the computer. Design the motion of the robotic arm and perform appropriate tests to verify the program's accuracy. When the system is running, manual or automatic operation commands are issued through the front panel of the labview. When the system is running automatically, the system writes the serial port data 1 and the arduino program determines that the operation is automatic. When writing manually, the serial port data is a string array. When the arduino program judges to be false, the data processing is performed. The processed data is written into each interface to control the rotation of each servo.

Arduino's build environment already contains the servo's control library file, so we can directly call the existing servo control function at the beginning of the program. The common statements of the servo function are as follows: attach -- set the servo interface; write -- set the servo rotation angle, the settable angle range is 0 ° to 180 °; read -- read servo angle; attached -- determine whether the servo parameters have been sent to the interface of the servo; detach -- to separate the servo and its interface, the interface can continue to be used as the common interface.

Firstly, define 6 servo angle variables, an integer variable (initialized to 0), an array of strings (stored angles sent by the upper computer), and a boolean variable, include the servo head file and define 6 servos. Initialize the serial port bit rate of 9600 and define the corresponding interface of each servo. According to the instructions of arduino, the servo interface is defined as 3, 5, 6, 9, 10, and 11

respectively. If there is data in the serial port, read the data and assign it to the variable data. If it is judged to be true, it will be automatically run, otherwise it will be run manually.

During manual operation, the maximum value of each servo angle is 180. The data of servos 1 to 6 are stored in the string received by the serial port. Firstly, it is determined whether the data meets the requirements. When the requirements are satisfied, the serial port data is assigned to each element of the array. In manual mode, the data received by the serial port is a character string, and the character 0 represents 48. Therefore, the serial port data needs to be processed. Each of the three elements of the array is the angle value of a servo, and the data is arranged in the order of the angles of the servos 1 to 6. Write each angle value to the corresponding interface of each servo and send the angle value from the serial port. Then clear the serial port, reset the variables and array elements for the next cycle. The servo is initialized to wait state. Read the current angle of the servo, and then use the map function to process the data and write it to each serial port. Desktop robot is shown in Figure 1.

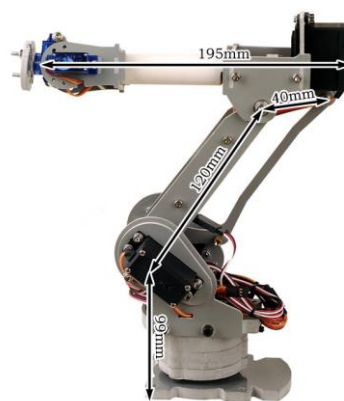


Fig. 1 Desktop robot

3. Labview programming

In this design, six knobs are placed on the front panel to input data for each of the six servos, and an automatic manual joystick and a stop button are also provided. The front panel of the arduino-based desktop robot control system is shown in Figure 2.

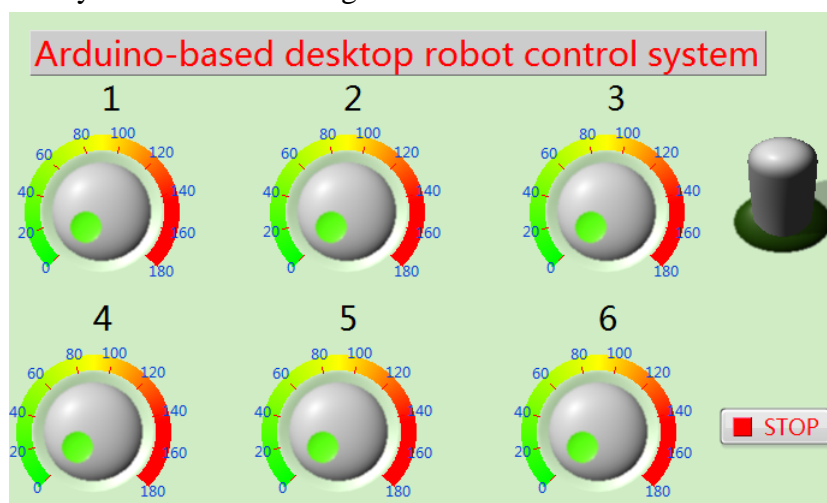


Fig. 2 The front panel of the arduino-based desktop robot control system

When the joystick hits the manual position and there is no wrong input, the angle data of each servo will be processed. When the servo input angle is a single digit x, the condition structure outputs the data 00x after processing through the connection string, as shown in Figure 3; When the servo input angle is a two-digit number xx, the condition structure output data 0xx, as shown in Figure 4; When

the input angle of the machine is three digits xxx, the condition structure outputs the original data, as shown in Figure 5. Firstly, convert the value to a decimal string and perform different conditional branches depending on the length of the string. Afterwards, the data of the 6 servos are written into the serial port as an array of strings through the shift register and the connection string .

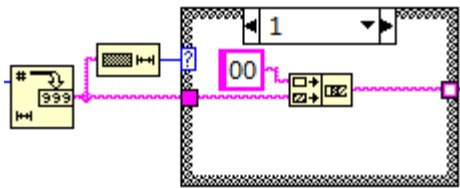


Figure. 3 Conditional branch 1

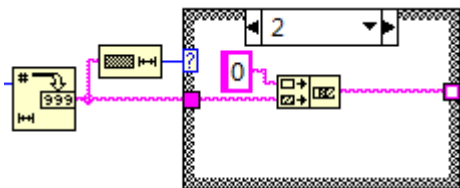


Figure. 4 Conditional branch 2

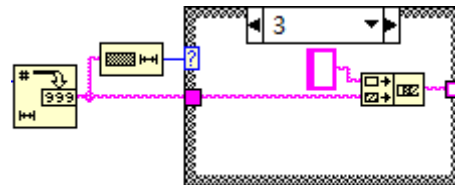


Figure. 5 Conditional branch 3

The labview block diagram of the control system includes a while loop, a for loop, three conditional structures, and six servo angle input controls. The specific block diagram is shown in Figure 6.

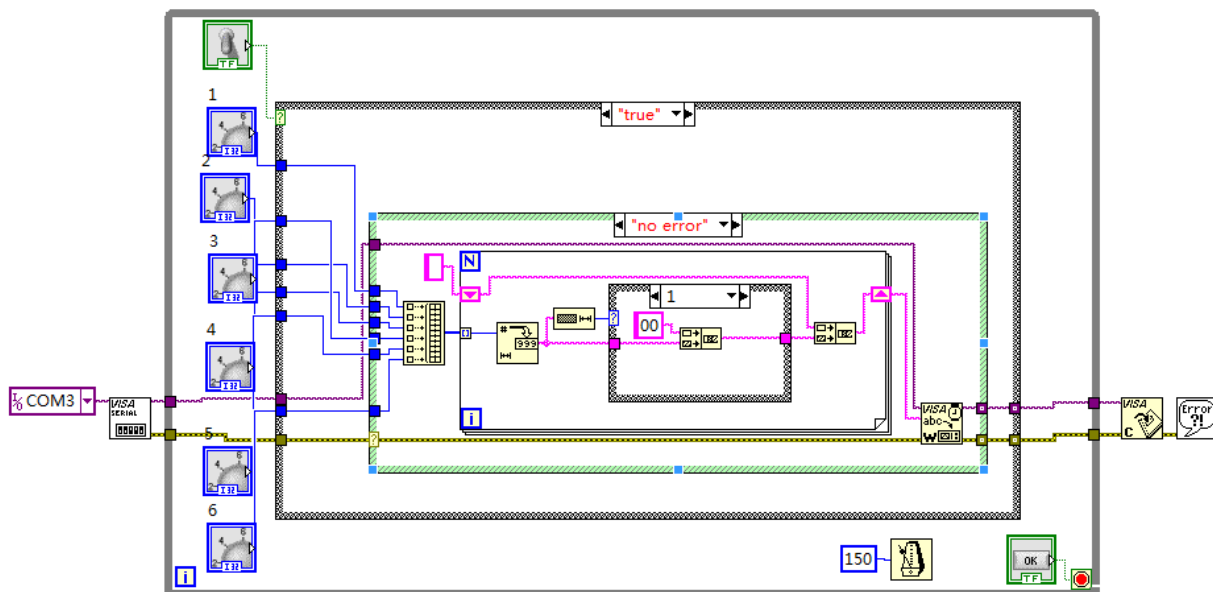


Figure. 6 The specific block diagram

The system operation includes the following situations: The joystick hits the automatic position and there is no wrong input, the joystick hits the automatic position and there is an error input, the joystick hits the manual position and there is an error input, the joystick hits the manual position, there is no error input and specify the rotation angle of each servo.

4. Conclusion

The arduino and labview programs were written separately, through the serial port communication between the two to achieve the arduino-based desktop robot control system design, to achieve rapid development of the host computer software and provide the ideas and theoretical models for the practical application of the industrial robot arm.

References

- [1] Ruiyan Cai. Principles and Applications of Arduino, Electronic Design Engineering, vol. 20(2012),155-157.
- [2] Li Zhang, Liang Cai. Design and Implementation of Virtual Signal Generator Based on LabVIEW, Foreign Electronic Measurement Technology, vol. 33 (2014), 82-85.
- [3] Honggang Li, Suping Zhang. Design of Multi-channel Data Acquisition System Based on MCU and LabVIEW, Foreign Electronic Measurement Technology, vol.33(2014), 62-67.
- [4] Information on <https://en.wikipedia.org/wiki/Automation>
- [5] Information on https://en.wikipedia.org/wiki/Robotic_arm