*2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)*
***IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore***
*DOI: 10.7763/IPCSIT.2012.V53.No.2.39*

# The Design and Implementation of the Wireless Remote Control Car

Dai Jinbo [1] , Wang Shaokun [2], Zhao Hongwei [1, 2+], Wang Xu [1]

Department of Computer Science and Technology [1], Department of Software [2], Jilin University, JLU

Changchun, China

**Abstract—**Due to the limitation of a system platform, in the process of the intelligent robots research, many graphics and image processing algorithms which designed for intelligent robots can only be emulated in the PC, but the practical effect of these algorithms can not be tested in a real robot platform. Build a wireless remote control car system as an experimental platform can solve this problem. We use an Atmega128 MCU to control the car's movement, and adopt a NRF24L01 chip as a transceiver to communicate with computer. By loading a C3088 digital camera and a gesture recognition test, the usability of this platform was eventually verified. However, the test result shows that the system still has much room for optimization in real time, wireless transmission distance and functional complexity.

**Keywords-** Intelligent Robot, Remote Controlling Car, Wireless Communication, Human-Computer Interaction, Gesture Recognition

## 1. Introduction

In recent years, intelligent robots have been a hot spot for researchers. Regarding the intelligent robot, visual technology theories based on the artificial intelligence model, the neural network model and the cognitive model emerge one after another incessantly. These techniques often require a lot of complex graphics and image processing algorithms or signal processing algorithms as the supplement. When an algorithm was proposed, verification in a real robots environment tends to be more convincing and more interesting. Meanwhile, the most important thing is that only in this way can we find the problems which never appear in the PC emulation. However, due to the lack of tools and technique, to confirm these algorithms directly in the robot's environment is actually difficult. For example, MATLAB can not be used in a robot platform. To solve this problem, design and build a wireless remote control car as a test platform from the aspects of functional requirements and system resource usage. The platform adopts a distributed structure, with an image sensor to detect the external environment, and a PC is used as the core of computing and decision-making. This keeps the image acquisition and partial pretreatment is processed on the remote control car, and the complex algorithms which need resources or special tools on PC are executed on the PC. The information interaction between PC and remote control cars is accomplished by wireless communication.

## 2. Framework of Wireless Remote Control Car

The overall design scheme of the system based on the software and hardware co-design theory which argued by Lagnese [1], from a functional information point of view, system is divided into three parts: car-end, PC-receiver-end and PC-control-end. As shown in Figure 1, system's three parts connect together through the different protocols, constitutes a complete system.

+ Corresponding author.
*E-mail address*: zhaohw@jlu.edu.cn

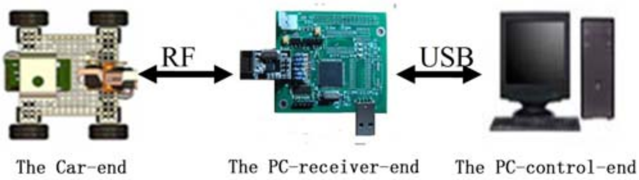



Fig.1. The wireless remote control car framework diagram.

### 2.1. The Car-end

The most left part in Figure 1 is the car-end. Its function is to control the electric motor, capture images, and sent the image to PC-receiver-end. According to the specific functions car-end can be subdivided into three modules: car control module, image acquisition module and Radio Frequency (RF) interface module. Image acquisition module and RF interface module are the components of the image acquisition terminal hardware circuit.

Car control module through the control of four DC motors with electronic brushes to achieve control operations moves the car, such as forward, backward, turn left, turn right and other operations. For each operation will have a time limitation, which lasts 1 to 3 seconds. Car control module receives the specific control commands which are made by the PC-control-end from the RF interface module through UART port, and then execute independently. Image acquisition module controls the image sensor to capture images. RF interface module is responsible for sending the images collected by Image acquisition module to PC-receiver-end through RF transmission and also responsible for sending commands to car control module from the UART port.

### 2.2. The PC-receiver-end

The middle part in Figure 1 is the PC-receiver-end, which has very similar functions with the RF interface module of car-end. It is mainly used to implement a wireless connection between the car-end and the PC-control-end. It receives the image data which collected by car-end and sends the control commands to remote control car through the RF interface. There are two interfaces designed for PC-receiver-end doing data exchange with PC-control-end: RS232 and USB. RS232 provides a slower speed, however, because it is relatively easy to develop and debug, this port was used for early system testing; The USB port provides a faster speed connection, finally it is determined to be the interface between PC-receiver-end and PC-control-end.

### 2.3. The PC-control-end

The most right part in Figure 1 is the PC-control-end, which consists of the PC host and peripheral, is decision-making and control center of the whole system. This part is responsible for displaying and processing image data and intelligent decision-making. If wants to receive and the transmission data through the USB connection, it must contain the USB driver which matches with PC-receiver-end. It receives the image data which the PC-receiver-end transmits. After adding the essential format information for these data, it becomes an image and the image will be shown to users. Meanwhile use the unique resources and tools which on PC system platform to analyze the received image data with the image processing methods which can not be implemented on car-end. PC-control-end program returns the parameters of processing result and generates commands which the wireless remote control car should execute. PC-receiver-end sends them to the wireless remote control car to achieve corresponding operations.

## 3. Hardware Architecture

System hardware consists of two main parts, image acquisition terminal hardware circuit and image receiving terminal hardware circuit. Fig. 2 shows the architecture of the image acquisition terminal hardware circuit, and figure 3 shows the architecture of the image receiving terminal hardware circuit.

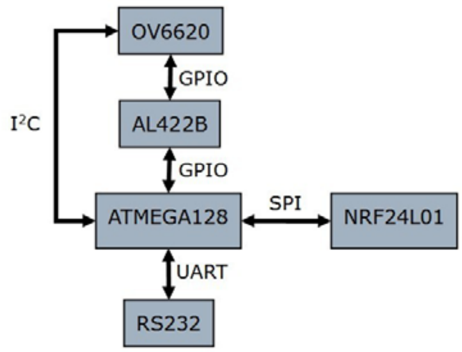



Fig.2. The image acquisition terminal framework diagram and the methods of chips connection.

## 3.1. The image acquisition terminal hardware circuit

The image acquisition terminal circuit installed in the car-end of the system, which is the physical layer structure of the car-end, the major function of the image acquisition terminal circuit is completing the acquisition of the car and communicate with the PC-control-end through radio frequency. Due to the moving range of the remote control car is large, the circuit needs power supply by battery pack. According to this fact, when choosing the components for the circuit and designing the circuit, we have to consider about power consumption, meanwhile in the low power loss situation must guarantee that the controlled distance of the car is as far as possible. The image acquisition terminal circuit use an Atmega128 chip as a MCU [2], which is high-performance, low power consumption and low cost 8-bit MCU, the framework adopt the RISC architecture, consists of 133 instructions, most instructions can be completed within a single clock cycle, avouching the control of the car-end is real-time and stability. Image collecting uses a C3088 digital camera, the C3088 adopts OV6620 chip [3], which is a low-end CMOS image sensor, with the maximum supported resolution 352×288. Since the speed of ATMEGA128 collection rate is far below the OV6620, meanwhile the RAM storage space is too small to store a full size picture, to solve this problem we set an AL422B between ATMEGA128 and OV66020 as a frame buffer [4]. AL422B is consists of 3M-bit DRAM, and configured as 393216-byte FIFO. Its interface is simple, and it has a high read-write speed. Because of the radio frequency chip need receiving command and delivering image at the same time, so the chip should support full-duplex communication mode. NRF24L01 chip is a transceiver based on the 2.4G base band transportation protocol [5], with the speed of 2Mbps, using NRF24L01 wireless communication module can not only ensure the speed, power consumption, but also the module have data link layer functions, such as data verification, error control, to simplify system design itself. RS232 provides a communication interface between the ATMEGA128 MCU and PC [6], as a platform to facilitate follow-up testing during the extended reservation.

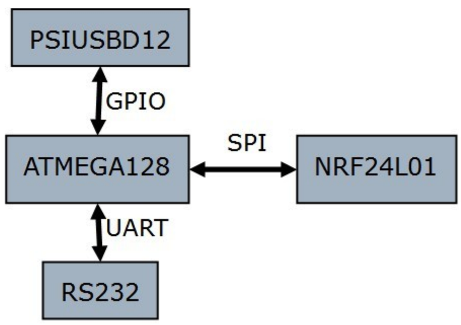



Fig.3.The image receiving terminal framework diagram and the method of chips connection.

## 3.2. The image receiving terminal hardware circuit

The image receiving terminal circuit is the primary part of PC-receiving-end. Both in structure or function, it is similar with the image acquisition terminal hardware circuit, and it is designed as a pluggable external device, by the way of the connection between USB interface and PC-control-end. ATMEGA128 is used as the same circuit as the MCU, NRF24L01 as a wireless communications chip. To guarantee the communication speed between the PC-receiver-end and the PC-control-end, the PDIUSBD12 is choose as the USB interface chip of the device [7].

The two above mentioned parts of the hardware circuits construct a complete picture sending and receiving system. Because ATMEGA128 only supports I2C, SPI and USART protocol, so in the realization, using some part of GPIO to simulate the corresponding timing to control some part of the devices, such as PDIUSBD12, AL422B and so on.

## 4. The Main Process Procedure

Since the system is divided into car-end, PC-receiver-end and PC-control-end, and the PC-control-end needs an additional USB driver to communicate with the PC-receiver-end, so need four separate programs to complete the whole system. Among them, the basic procedure of car-end and the PC-receiver-end is very similar. We presented car-end and PC-control-end program flow diagram here.

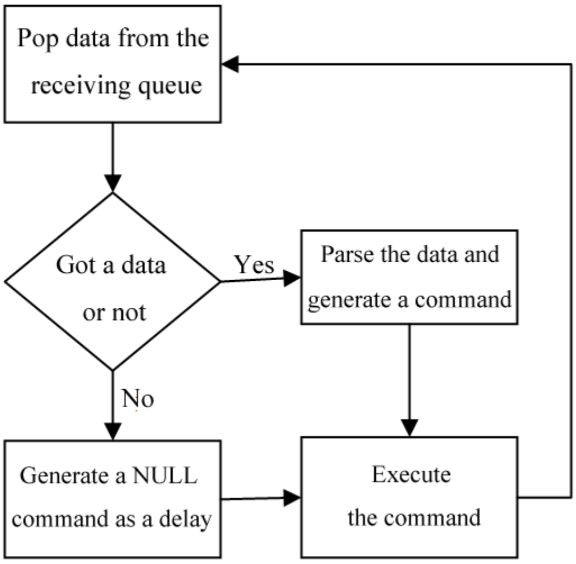



Fig.4. The car-end process flow chart

Fig. 4 shows the procedure of the car-end. As shown in figure, the program's main loop is a commands parse and a commands response process. Command data is received into the receiving queue through interrupt response of receiving process. If got no data from the queue, then execute the null command as a delay. If received a data from the queue, then get the command data from receiving queue, and parse it. Execute the command which has been parsed.

PC-control-end is the core of the entire system. It is the command center of the system and the origin of system action. Fig. 5 shows specific process. Thus it can be seen that the main logic of PC-control-end is a loop operation. In the cycle, the image acquisition, image display, image analysis and the operation of sending the analytic result are accomplished.

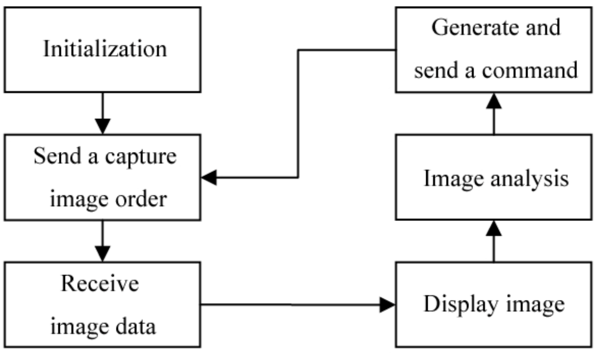



Fig.5. The PC-control-end process flow chart

## 5. System Tests

After the wireless remote control car platform has been built, we carried out the connection speed test, the wireless communication speed test, the range of wireless communication test and other basic performance tests to the system. In order to verify the design of remote control car platform to address the issue which proposed at the beginning is feasible, we design a Gesture Recognition Experiment to prove that the platform can verify graphics processing algorithms using PC unique software in a real robot environment. In Gesture Recognition Experiment we control the remote control car driving directions by changing gestures and observe the result of the wireless remote control car's determination of gestures. The testing process involves image acquisition test, image de-noising algorithm test, image edge detection algorithm test, gesture extraction algorithm test and gesture recognition test [8]. The test PC-control-end has an Intel Pentium4 1.6GHz CPU and 448MB memory. The test operating system is Microsoft Windows XP and the testing software program is compiled with the visual C++6.0.

### 5.1. RS232 port speed test

TABLE I. RS232 SPEED TEST TABLE

| **Number** | **Test items** | | | |
|---|---|---|---|---|
| | ***Read Time*** | ***Write Time*** | ***Read Speed*** | ***Write Speed*** |
| 1 | 9.125S | 8.781S | 11.11K/S | 11.54K/S |
| 2 | 9.110S | 8.781S | 11.13K/S | 11.54K/S |
| 3 | 9.125S | 8.781S | 11.11K/S | 11.54K/S |
| 4 | 9.109S | 8.782S | 11.13K/S | 11.54K/S |
| Average | 9.124S | 8.781S | 11.11K/S | 11.54K/S |

Test Method: Set the baud rate 115200, send 101376 bytes, and use the function clock () to calculate the time. This function is used to get the current process has past how many clock ticks from the very beginning. Clock ticks number divided by CLOCKS_PER_SEC can obtain a communication time (in millisecond). Table 1 shows four randomly selected test results from 100 times tests and the average communication speed.

### 5.2. USB speed test

Test Method: After the initialization, receives 101376 bytes data from the USB port. After receiving the data, does not make any processing and drops them directly. Writing speed test does in the same way. Similarly, use clock () function to measure time. Table 2 shows 4 randomly selected test results from 100 times tests and the average communication speed.

TABLE II. USB SPEED TEST TABLE

| **Number** | **Test items** | | | |
|---|---|---|---|---|
| | ***Read Time*** | ***Write Time*** | ***Read Speed*** | ***Write Speed*** |
| 1 | 0.797S | 0.891S | 127.20K/S | 113.78K/S |
| 2 | 0.801S | 0.891S | 126.56K/S | 113.78K/S |
| 3 | 0.782S | 0.890S | 129.64K/S | 113.91K/S |
| 4 | 0.782S | 0.890S | 129.64K/S | 113.91K/S |
| Average | 0.790S | 0.890S | 125.40K/S | 111.17K/S |

### 5.3. Wireless transmission speed test

Test Method: After initialization, the sending end sends 101 376 bytes data, and the receiver receives. Using the clock () function to measure the time on PC. Before the sending end starts to send data, it sends a signal bit to PC from RS232 port to inform PC the transmission start. Ignore the one bit sending time from

RS232 to PC. The receiver speed tests in the same way. Table 3 shows 4 randomly selected test results from 100 times tests and the average communication speed.

TABLE III. WIRELESS TRANSMISSION SPEED TEST TABLE

| Number | Test items | | | |
|---|---|---|---|---|
| | ***Receiving Time*** | ***Writing Time*** | ***Receiving Speed*** | ***Writing Speed*** |
| 1 | 2.353S | 2.354S | 43.08K/S | 43.07K/S |
| 2 | 2.357S | 2.353S | 43.01K/S | 43.08K/S |
| 3 | 2.353S | 2.357S | 43.08K/S | 43.01K/S |
| 4 | 2.353S | 2.397S | 43.08K/S | 42.30K/S |
| Average | 2.354S | 2.365S | 43.07K/S | 42.87K/S |

## 5.4. Wireless communication range test

Test in open space and no barriers, the wireless remote control car controlled distance may up to 18.2 meters. In the case of barriers, controlled distance is about 7 meters (depend on the barriers). For giant shields, remote control car is impossible to communicate with PC-control-end.

## 5.5. Gesture recognition experiment

*1) Image de-noising algorithm test*

Fig. 6 (a) shows the collected image data is processed as the gray scale image in PC. As the packets lost during transmission, so the image appears a number of distinct horizontal stripes. Fig.6 (b) shows the result of median filter using 3×3 filter window [9]. Image de-noising uses 0.23 seconds in average.

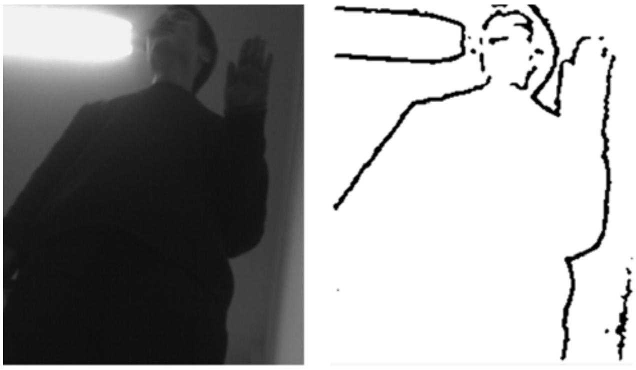
(a) The gray scale image. (b) The result of median filter.

Fig.6. The image de-noising comparison diagram

*2) Image edge detection algorithm test*

CANNY edge detection algorithm is a widely used efficient method [10], it is roughly divided into four steps: Smooth image and reduce the noise with Gaussian filter; Calculate gradient amplitude and direction by finite difference; Perform non-maxima suppression to gradient amplitude; Examine and connect the edge with the double-threshold method.

After several experiments, we found that use first-order partial derivatives and gradient non-maxima suppression method for edge detection is ineffective for this experiment, but more complicated calculation., we modify the traditional CANNY edge detection algorithm: First calculate the edge using the Laplacian-Gauss operator. Then do the CANNY algorithm's fourth step, namely dual-threshold detection and edge joint. Profit from the thought of the image corrodes, performing median operation to edge image with a 3×3 window.



(a) The original image. (b) The edge image.

Fig.7. The effect picture of edge detection

Fig. 7 (a) shows the original image and (b) is the effect picture of edge detection algorithm used in this article. Edge threshold value is 150, and the average time for the edge detection is 0.15 seconds.

*3) Gesture extraction test*

Here, we use the classical threshold method [11], in order to extract the binary image of the gesture area. In Figure 8, on the left is the original image, and the right part is the binary image which processed with regional threshold. Gestures extract uses 0.015 seconds in average. In 100 different images tests, in 26 of them the sign area can not be found. Among the 26 images there are 8 images shows a greater tilt gesture.

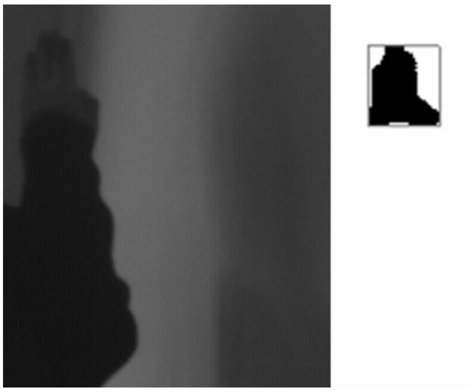


Fig.8. The effect picture of gesture extraction

*4) Gesture recognition test*

After obtaining the gesture region, needs to analyze this gesture region to distinguish the specific hand signal. As the purpose of this study was to verify the feasibility of remote control car platform, only establishes the recognition pro and con two simple hand signals. In order to simplify the design, we calculate areal coordinate with the mount about the center of gravity to recognition hand signal. If the center of gravity abscissa value is bigger than the gesture window central point abscissa value, then can determine the thumb appears in the left side of the window, and generate a command to make remote control car walking forward. On the contrary, can determine the thumb appears in the right side of the window, and generate a command to make remote control car walking backward.

TABLE IV. RESULT OF GESTURE RECOGNITION TESTS

| **Number** | **Test items** | |
|---|---|---|
| | ***Time spent*** | ***Recognition mark*** |
| 1 | 2.925S | 1 |
| 2 | 3.105S | 0 |
| 3 | 2.901S | 1 |
| 4 | 3.019S | 1 |
| Average | 3.021S | 0.69 |

Repeatedly carries out the gesture recognition test 100 times, the correct recognition marks 1, otherwise marks 0. As shown in Table 4, 69 pictures have been distinguished correctly. The average recognition time is 3.021 seconds.

## 6. Conclusion

This paper implemented and verified the remote control car loads a camera, transfers image information by RF modules, processes the image data by PC and return the decision information back to the remote control car, namely, the wireless remote control car system platform. This platform solves the problem effectively that many graphic and image process algorithms in intelligent robot domain can only be emulated in PC but unable to be used in a real robot environment. And PC machine's strong handling ability and the rich software and hardware resources can be used for making up the limited processing ability of robot platform, while simplifying the underlying hardware design of the robot. The wireless remote control car platform is designed with low power consumption, low cost, easily implemented, practical and high scalability. However, the results of the experiment data shows that the system still has much room for optimization in real time, wireless transmission distance and functional complexity so that it can be applied to the algorithm validations which need more requesting and more demanding conditions.

## 7. References


[1] E. D. Lagnese and D. E. Thomas, "Architectural partitioning for system level design," in Proc. 26th DAC. June 1989, pp. 62-67.

[2] Atmega128 Datasheet [EB/OL]. http://www.datasheetcatalog.com/datasheets_pdf/A/T/M/E/ATMEGA128.shtml.

[3] OV6620 Datasheet [EB/OL]. http://www.cs.cmu.edu/~cmucam/Downloads/ov6620DSLF.PDF.

[4] AL422B Datasheet [EB/OL]. http://www.datasheetsite.com/datasheet/AL422B.

[5] NRF24L01 Datasheet [EB/OL]. http://www.nordicsemi.com/index.cfm?obj=product&act=display&pro=89.

[6] MAX232 Datasheet [EB/OL]. http://rocky.digikey.com/WebLib/Texas%20Instruments/Web%20data/MAX232,232I.pdf.

[7] PDIUSBD12 Datasheet [EB/OL]. http://www.waveshare.net/Datasheet_PDF/PDIUSBD12-PDF.html.

[8] Bulzacki, A., Lian Zhao, Ling Guan and Raahemifar, K., Computerized Recognition of Human Gestures[C]. Electrical and Computer Engineering, 2008, CCECE 2008 Canadian Conference, 2008: 1003-1006

[9] Liu Limei, Sun Yurong and Li Li, Research of median filter technology, Journal of Yunnan Normal University, Vol. 24, No.1, pp. 23-27, Jan. 2004.

[10] John Canny. A computational approach to Edge Detection[C]. Pattern Analysis and Machine Intelligence, 1986, 8(1): 679-697

[11] HAN Si-qi, WANG Lei, A Survey of Thresholding Methods for Image Segmentation, System Engineering and Electronics, Vol.24, No.6, pp. 91-94, 2002.