The design of data acquisition system of program automatic shelling based on ARM

Ren Yan-yan[1],CUI Rong-rong[2]

{wudan22@tom.com[1], nide25251@tom.com[2]}

[1,2]College of Mathematics and Computer Science, Chifeng University,Chifeng 024000, China

Abstract. In view of the low efficiency and poor effect of the current program automatic shelling data acquisition system, the design of the program automatic shelling data acquisition system based on arm is proposed. Its hardware consists of ARM (Advanced RISC Machines) processor, ads830 collector and interface conversion module; the software design includes main control software, GPS (Global Positioning System) acquisition data receiving module, FIFO (Flight Inspection Field Office) data cache module and local data processing module. Through the contrast experiment, it is proved that the designed program automatic shelling data acquisition system not only plays an excellent role in shelling effect performance, but also far better than the traditional design in data acquisition efficiency, and higher than the standard line, with high acquisition efficiency and stability.

Keywords: ARM; Program automatic shelling; Data acquisition;

1 Introduction

With the development of computer technology, arm has become one of the focuses of the current IT (Information Technology) industry, showing a huge market demand. The core technology of embedded system is embedded processor and embedded operating system. ARM company's 32-bit RISC (Reduced instruction set chip) processor, with its high speed, low power consumption, low cost, strong function, unique 16 / 32-bit dual instruction set and many other excellent performance, has become the first choice processor in mobile communication, handheld computing, multimedia digital consumption and other embedded solutions [1]. Among all kinds of embedded operating systems, Linux is widely used in data acquisition, instrumentation, measurement and control system, handheld devices and other embedded system applications because of its clear structure and open code.

In recent years, malicious codes (such as viruses, Trojans, worms, etc.) generally use some technologies with strong survivability and high protection strength to hide and transform, so as to avoid the scanning and analysis of detection tools (such as anti-virus software), and the technology of adding shell is a typical representative of them [2-3]. According to statistics, in 2003, nearly 29% of malicious code was cased, which rose to 35%

EAI MOBIMEDIA 2020, August 27-28, Harbin, People's Republic of China

DOI 10.4108/eai.27-8-2020.2294619

in 2005 and exceeded 80% in 2007. The development trend of malicious code brings great challenges to the detection tools. After the program is shelled, only the part of the program that is shelled is exposed, which generally does not contain malicious intention. After the content of the original program is compressed and encrypted, the detection tools have been difficult to identify, and the detection success rate is very low. Therefore, how to restore the content of the program and obtain the execution behavior of the program is the focus of malicious code detection technology research. At present, there are two kinds of Shelling procedures: manual shelling and automatic shelling. Manual shelling requires professional reverse engineering experience of analysts, and the process is tedious and energy-consuming [4-5]. Automatic shelling often depends on some existing features, with poor generality, especially for some non-public shell professional analysts who often spend a lot of time to analyze. Moreover, with the development of shelling technology, malicious code not only uses one shell, but also can have multiple shells, which brings great difficulties to the analysis of malicious code. Because the traditional manual and directional shelling methods have obvious defects, such as the lack of universality, the difficulty to keep up with the progress of shelling technology, the development speed of shelling cases, the need to spend a lot of manpower and material resources, so researchers are committed to the research of automatic shelling technology. At present, there have been some new research results in the field of program automatic shelling, such as polyupack, renovo, Ma lwarenorma liza, etc., but these technologies can not evaluate the effectiveness of extracting data. In the face of some highly protective technologies, it is difficult to accurately recover the program content strength, such as stone byte and virtual machine technology. Based on this, it is necessary to optimize and innovate the traditional automatic shelling data acquisition system.

2 Hardware composition of automatic shelling data acquisition system based on ARM

2.1 Selection of ARM Processor

In a multitasking system, the kernel is responsible for managing each task, or allocating CPU to each task. The first step of setting up the hardware development platform of data acquisition system is to make a good choice of ARM core. Various series of arm architecture, such as ARM7, armg, armge, arm10e, securcore, xseale of Intel, StrongArm of hitel, etc., should consider the following main factors when selecting different embedded system applications:

For the selection of mmij unit, if you want to use wince or Linux and other operating systems to reduce the software development time, you need to select ARM chip with MMU (Memory Management Unit) function above arm720t, arm720t, StrongArm, arm92ot, arm922t, arm946t with MMU function. ARM7TDMI does not have MMU, and does not support Windows CE and most Linux.

The system clock controller determines the processing speed of ARM chip. The processing speed of ARM7 is 0.gmips/mhz, the common main clock of ARM7 is 20mhz-133mhz, the processing speed of arm is 1.1mips/mhz, the common main clock of arm system is 100mhz-233mhz, and the maximum of arm10 can reach 700MHz.

Internal memory capacity, when large capacity memory is not needed, ARM chip with internal memory can be considered.

USB (General serial bus) interface, most arm chips have USB controller, some even have USB host and USB slave controller at the same time.

Among the instructions provided by some chip suppliers, the number of IO (Incoming Orders) usually indicates the maximum possible number of GPIO (General I / O port), but many pins are multiplexed with address line, data line, serial port line and other pins. In this way, we need to calculate the actual number of gpios that can be used in the system design.

The interrupt controller and arm core only provide two interrupt vectors: fast interrupt (FIQ) and standard interrupt (IRQ). However, different semiconductor manufacturers have added their own interrupt controllers in chip design to support hardware interrupts such as serial port, external interrupt and clock interrupt. External interrupt control is an important factor in chip selection. Reasonable external interrupt design can greatly reduce the workload of task scheduling. For example, for Philips saa7750, all gpios can be set to FIQ, and four interrupt modes can be selected: rising edge, falling edge, high level and low level.

LCD (Liquid crystal display) controller, some ARM chip built-in LCD controller, some even built-in 64K color TFT (Thin film transistor) LCD controller. In the design of PDA (Personal digital assistant) and hand-held display and recording equipment, ARM chip with built-in LCD controller, such as 53c2410xis more suitable. As shown in figure 1:

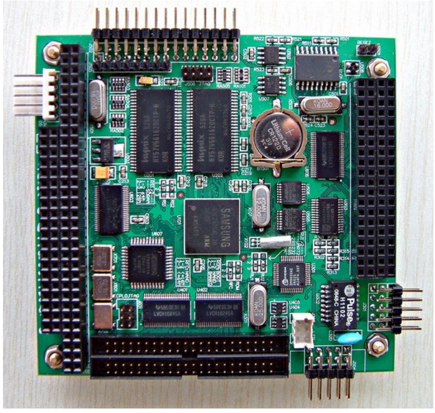


Fig. 1. 53C2410X ARM processor

ADC (Analog to digital converter) and DAC (Damage Assessment Center). Some arm chips have 2-8 channels and 8-12 bit general-purpose ADC, which can be used for battery detection, touch screen and temperature monitoring. Philips saa7750 also has a built-in 16 bit stereo audio ADC and DAC with headphone driver.

Expansion bus: most arm chips have external SDRAM (Synchronous dynamic random access memory) and SRAM (Static random access memory) expansion interfaces. The number of chips that can be expanded by different arm chips is different, that is, the number of chip selection lines. The external data bus has 8 bits, 16 bits or 32 bits.

UART (Universal asynchronous transceiver) and IrDA (Infrared data communication), almost all arm chips have one or three UART interfaces, which can be used to communicate with PC or debug with angel.

DMA (Direct memory access) controller, some arm chips are integrated with DMA, which can exchange data with external devices such as hard disk at high speed, and reduce the CPU (central processing unit) resource occupation during data exchange. ARM series microprocessors provide the best performance in terms of high performance and low power consumption. Arm series microprocessors have the following features: 5-level integer pipeline, higher support for instruction execution efficiency; 32-bit arm instruction set and 16 bit thumb instruction set; support for high-speed AMBA bus interface; full performance MMU (Memory Management Unit), support for Windows CE (chief engineer), Linux, palm 05 and other mainstream embedded operating systems; MPU supports real-time operating system; support for data cache and instruction cache, with higher instruction and Data processing capacity. As shown in figure 2:

Fig. 2. ARM series microprocessors

2.2 ADS830 collector

The ads830 high-speed acquisition chip with a sampling rate of 10ksa / S ~ 60msa / s and a bit of 8 bits is selected from TI company. Analog signal input amplitude of the module is ±

5V, digital interface level is 3.3V, module power supply is + 5V single power supply, analog signal input interface adopts two forms, one is pin interface, the other is SMA interface, SMA interface is used in this design. The ADS830 hardware structure is shown in Figure 3.

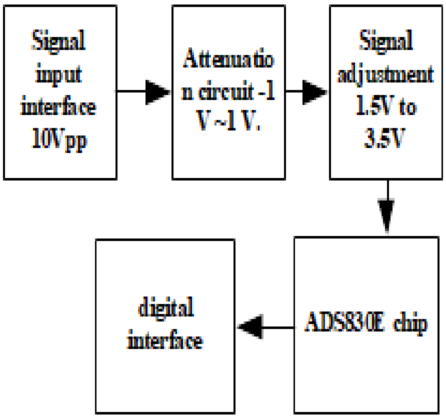
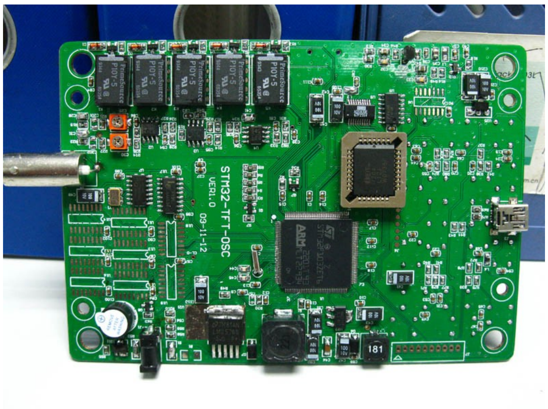

Signal input interface 10Vpp
Attenuation circuit -1 V~1 V.
Signal adjustment 1.5V to 3.5V
ADS830E chip
digital interface

Fig. 3. ADS830 Hardware Structure

In the signal conditioning part, the input signal range - 5V - + 5V is replaced by 1.5v-3.5v to meet the input range of ADC (Analog to digital converter) data acquisition; ad data acquisition adopts single terminal input with a reference voltage of 2V, i.e. the input voltage range is 1.5v-3.5v, and the digital interface adopts 3.3V logic level to facilitate direct connection with FPGA (Field Programmable Gate Array). The output interface of ads830 module is two rows of 5-pin, 10 interfaces in total, which are D8-D1, CLK and N / C (not used). D8-D1 is the digital level of output, which is connected with FPGA / 0 interface and can transmit the collected data to FPGA. CLK is controlled by CLK clock accessed from external fpgai / O. The ADS830 collector is shown in Figure 4:



兘**Fig. 4.** ADS830 collector

2.3 Interface Switching Module

The output of ads830 high-speed data acquisition module is 10PIN female seat, with two rows of ten pins output, while the available interface of fpgai / 0 is one row of 16 pins. If the interface conversion is not carried out, it can only be connected with DuPont wire. For high-speed data, the connection of DuPont wire may cause data delay and loss, resulting in inaccurate data. Therefore, the interface conversion module is made to complete ads83 Interface conversion from 0 to fpgai / O. In the process of making interface conversion module, because ads830 is a high-speed data acquisition module, the frequency of data acquisition is high and the change is fast, the distance and width of internal wiring can lead to inaccurate data [6]. The physical connection between ads830 and interface conversion module is roughly designed as follows: ads830e collector on the left side, interface conversion module on the right side, pin connection between P1 interface of interface conversion module and ads830, connection between P2 and fpgai / O of three core development board, transfer the collected data to FPGA, D8-D1 interface of P2 output is used for data reception, and interface CLK is used when FPGA provides ads830 Clock signal.

3 Software design of automatic dehulling data acquisition system based on ARM

3.1 Main control software

The main control software runs on the main control board, which is the core of the whole system. Development environment: take a computer installed Ubuntu 2.04 desktop system as the host development environment, port the main control board of Linux 3.2.0 system through network cable connection, and use NFS to mount the shared directory. Qt4.7 development tool is installed in the host computer. Arm none linumx gnueabi-g + + tool chain is used to compile the program. The program is copied to the main control board through the shared directory and debugged on the main control board. The main control software consists of three subprograms: main program, ad subprogram, parameter subprogram and communication subprogram.

The main program is responsible for completing the logic control and real-time detecting whether there is new parameter input and setting [7], configuration of communication module parameters, calling ad subroutine at the set sampling time, calling communication subroutine at the set sending time, etc;

The parameter subroutine establishes the network connection with the upper computer software, obtains the set parameters in real time, and saves them in the SD card;

The ad subroutine controls the SD card on the main control board;

The communication subroutine receives the data from the communication module, analyzes the GPS time and completes the time calibration.

3.2 GPS Data Acquisition Module

The difference between GPS receiving module and AIS receiving module is only that the ad_data [7:0] is replaced by CP, SGN and MAG, and the output part of ad_clk is removed. GPS data acquisition has been completed by internal ad. the three important data pins output by GPS chip are CP, SGN and MAG signals. GPS intermediate frequency data is sampled and converted at the rising edge of CP and read out at the falling edge of CP. therefore, the falling edge of CP should be judged in the program, and then read out the data. Because the receiving program of GPS data is based on a1s program, the rate cannot exceed 11.2m * 8bit/s. The sampling rate of CP is nearly 16.4m, which is higher than that of AIS. Therefore, the two bits of data should be folded three times in one byte, that is, {sgn1 、 Magi 、 sgn2 、 mag2 、 sgn3 、 MAG3 、 sgn4 、 mag4} to represent the first, second, third and fourth sampling points. The state of CP is recorded with the shift register busy reg, which is represented by the busy signal (busy reg). The falling edge trigger signal is generated as the trigger signal of data reception. In this way, there will be a delay of about 10ns in the trigger of K edge drop, but the operation of state machine can completely avoid this disadvantage and complete the data storage.

3.3 FIFO data caching module

FIFO is a first in, first out data cache array. Unlike general memory, its external output has no address line, so its advantages and disadvantages are obvious. Its advantages are convenient for user operation. Its disadvantages are that it can only store and take out data in sequence, and it can't write and read a certain data randomly according to the wishes of its own program [8]. It can complete the data transmission in different clock domain, complete the conversion transmission of data in low frequency domain and high frequency domain; it can complete the data buffer in different bit width, and the read and write clocks can be independent of each other. In FPGA program, FIFO IP core designed by Xilinx is used to complete data cache.

Set Key Parameters:

Width of FIFO: the number of data bits of a read / write operation of FIFO, which is 8 bits;

Depth of FIFO: how many n-bit data can be stored in FIFO (n is the width of FIFO), n is 1024;

The read clock is the UPP read clock, i.e. 111mhz;

The write clock is 100MHz; it is configured as normal type, and the read and write clock is in asynchronous mode.

FIFO through these configurations and pin connections, when the ad channel data write flag is set, the data is written into FIFO until 1024 bytes are written. At this time, the full flag signal is set, and the data is sent to the UPP module according to the UPP read clock, so as to complete the data cache and transmission.

3.4 Local Data Processing Module

Linux is a mature free operating system, supporting a variety of hardware platforms, fully compatible with POSIX 1.0 standard, with complete development community support [9], supporting multi-threaded and real-time processing. The local data processing function module can be divided into two processes: basic data processing function module and total data processing function module, as shown in Figure 5.

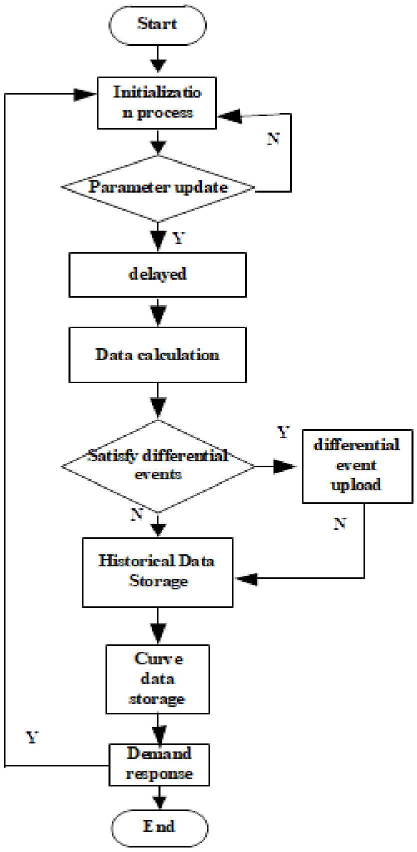


Fig. 5. Local data processing processes

The basic data processing module is responsible for analyzing the abnormal data of the measurement points and generating corresponding events. After the operation of the module, the message queue, measurement point parameters, shared memory, task parameters, etc. will be initialized and enter the main loop. After a certain delay, read the data and analyze whether it meets the event conditions. If the conditions are met, upload the data to the data communication module, read the statistical data of the terminal, and compare and analyze whether there is any change. If the data changes, read the historical data record and send it to the remote communication module [10], and then upload it to the main module Station. After all data processing is completed in the whole cycle, check whether the parameters of measurement point, event condition and task cycle have changed. If they change, carry out initialization again, and then continue data processing in the next cycle. The data processing function module of the total plus group is to make statistics on the data of the total plus group and send the data after the total plus to the data storage module for storage. After the operation of the module, the message queue, measurement point parameters, shared memory, total parameters, etc. will be initialized, and enter the main cycle, waiting for a total data statistics cycle. After that, the total data of the current data of the measuring point is calculated by using the summation formula, and the data is analyzed and compared to determine whether it meets the conditions of the difference situation. If it meets the conditions, a differential event will occur, and it will be sent to the remote communication module, and then uploaded to the main station by the module. Then, the current data and historical data records of all measurement points are generated into curve data and sent to the data storage module for storage. After all the processing of this cycle is completed, the parameters of each measuring point, the total plus parameters, the working cycle and the conditions of differential events are detected to determine whether these parameters have changed. If there is any change, restart and initialize, and then carry out the data processing of the next cycle.

4 Testing experiment

In order to verify the performance of the system (shelling effect and collection efficiency), six kinds of commonly used shelling data collection systems were compared and tested. Through comparison, the hypothesis of the experiment is verified. The experimental environment is shown in Table 1.

Table 1 Configuration information of development environment

Name	Configuration
Operating system	Microsoft Windows XP
Processor	Intel(R)Celeron(R) 2.6GHz

Internal storage	24.0 GB
Hard disk	8.0 GB
Database management software	Microsoft SQL server 2010 R2
Mathematical software	MATLAB

4.1 Experimental comparative analysis

In order to verify the effectiveness of the system, six kinds of commonly used shell tools are selected as the test object, and the remote control Trojan x-door is selected as the original program before shell adding. The six most commonly used shell adding tools are upx, pecompact, winupack, aspack, asprotect and mew. The experimental method is comparative experiment. The experiment verification process is completed under the control of other variables except the experimental variation. Figure 6 shows the data transmission terminal of program automatic shelling.

Fig. 6. Automatic dehulling of program data transfer terminal

Under the data transmission terminal, the remote service is used to collect the data in the program automatic shelling database.

4.2 Experimental result

In the actual test, the shelling effect of the system is compared with that of ployunpack, and the results are shown in Table 2.

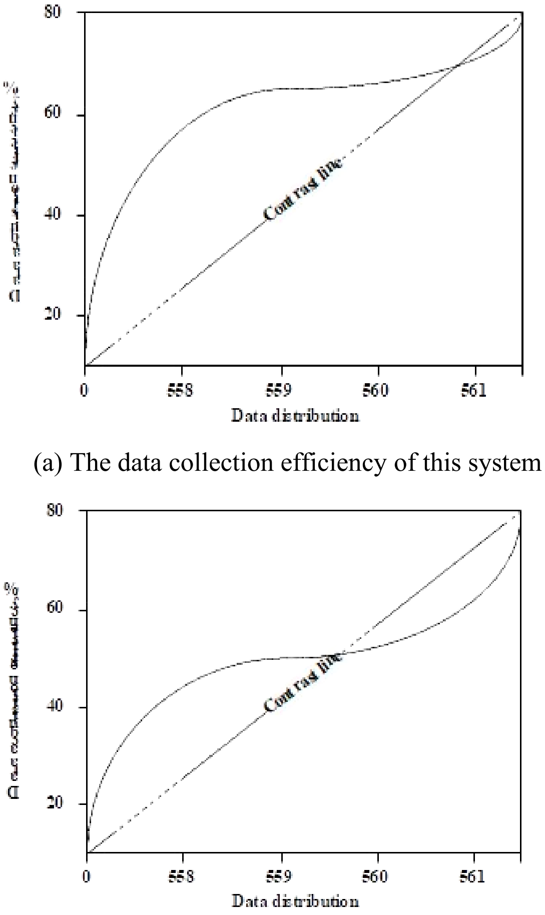
Table 2 Comparison of system shelling and polyunpack shelling in this paper

Shelling Tool	PloyUnpack	Dynamic dehulling
UPX	Yes	Yes
PECompact	No	Yes
WinUPack	Part	Yes
ASPack	Part	Yes
ASProtect	Yes	Part
MEW	No	Yes

Among them, yes indicates correct shelling, no indicates no shelling, part indicates incorrect entry point identification, but part of the original program is obtained.

From table 2, it can be concluded that ployunpack has not completely shelled except for the correct shelling of upx and asprotec. The shelling effect of the system technology proposed in this paper is better than ployunpack. At the same time, through the comparison of shelling time, the average time of shelling implementation in this paper is less than ployunpack. The reasons are as follows: first, ployunpack's single step comparison object is the generated decompilation result, while it is difficult to ensure that the binary executable is completely correct in the decompilation, and the shelling program often has the phenomenon of multiple shells. Compared with the system designed in this paper, ployunpac's processing effect on multiple shells is relatively poor; Second, ployunpac is based on single-step execution. Before each instruction is executed, it is checked and processed. The execution time of ployunpac is tens to tens of times of that of the original program, so the actual execution time of ployunpac is longer than that of the system in this paper.

In the actual test, the data collection efficiency of this system is compared with that of ployunpack, and the result is shown in Figure 7.



(a) The data collection efficiency of this system

(b) Data collection efficiency of traditional system

Fig. 7. Comparison of acquisition efficiency

According to figure 7, the design system in this paper is far better than the traditional design in data collection efficiency, and higher than the standard line, with high collection efficiency and stability.

5 Concluding remarks

In this paper, a program auto shelling data acquisition system based on arm is designed. By comparing with six traditional auto shelling data acquisition systems, it is proved that the system designed in this paper has high shelling effect, high efficiency and better stability in data acquisition.

References

[1] He, Z. Guangyou, Y. Xuehai, C.: Design of Remote Monitoring Terminal Based on Embedded

System OpenWRT. Journal of Hubei University of Technology, Vol. 33, no. 1, pp. 31-37 (2018)

[2] Lei, Z.: Research on Intelligent Detection System of Communication Power Based on Internet of Things. Intelligent Building and Smart City, no. 02, pp. 51-52 (2018)

[3]Jianxin, B. Guoping, C. Shangzhi,.X. et al.: Design of Electric Energy Data Acquisition Terminal Based on ARM. Agricultural Equipment and Vehicle Engineering, Vol. 56, no. 2, pp. 60-63 (2018)

[4] Jianjian,.Y. Wushan, C..: Communication design of embedded control system with multi - control fusion. Computer Measurement and Control, Vol. 26, no. 1, pp. 92-9498 (2018)

[5]Channing, H. Lina, H. Panpan, W.: Design of Real - time Monitoring System for Embedded Web Server. Application of Single Chip Microcomputer and Embedded System, Vol. 18, no. 3, pp. 46-51 (2018)

[6]Fuguo, C. Yujie, D. Ruimin, Z. et al.: Design and Implementation of Intelligent Substation Monitoring IED Based on ARM Platform. Instrument Technology and Sensor, Vol. 18, no. 4, pp. 28-31 35 (2018)

[7]Yewoo, O.: Design and implementation of wireless data transmission system based on ARM.China New Communications, Vol. 20, no. 7, pp. 74-74 (2018)

[8]Yan, L. Fengchen, H. Xijun, Y.: Design of Internet of things gateway based on ARM 11[J]. Microcomputer Applications, Vol. 34, no. 5, pp. 40-43+49 (2018)

[9]Yanjing, O.: Design of Time Synchronization Data Acquisition and Remote Management System Based on ARM. Telecommunication Technology, Vol. 12, no. 4, pp. 35-38 (2018)

[10] Rudder, C. Yungkeung, .: Design of Remote Data Acquisition System Based on Embedded Web Server and ZigBee. Journal of Tangshan University, Vol. 31, no. 3, pp. 7-10+19 (2018)