

## Data Acquisition System for Press Machine

Ku Xiangchen<sup>+</sup>, Li Jishun, Ma Wei, Xue Yujun

School of mechatronics engineering  
Henan University of Science & Technology  
Luoyang, P.R.China

**Abstract.** According to the actual production needs, the data acquisition system based on RS-485 is developed and used to detect the hydraulic pressure and slider displacement for 2500 ton press machine. The paper describes the composition of the hardware and the communication protocol adopted by the system. The emphasis is on how to realize the communication between the different modules. The paper also presents some key technologies of the system and its program method on BCB platform. The application result indicates that the system can meet the requirement of real time and high precision data acquisition in the process of press working.

**Keywords:** Data acquisition; serial communication; RS-485; C++ Builder

### 1. Introduction

The press machine exerts pressure to work piece and cause it deformation accordingly. Finally, the required work piece shape can be obtained. In the process of production or test, in order to improve the quality of parts and understand the deformation characteristics of materials, it is necessary to measure the pressure which the part receives and deformation displacement which the part produces. Therefore, the paper developed the data acquisition system of pressure and displacement for a 2500 ton four column press machine. The data collected and transmitted to the PC through RS-485 bus. RS-485 bus has strong anti-interference capability and suitable for long distance transmission. Borland C++ Builder 6.0 is employed to realize real time data communication between different modules of the system.

### 2. Hardware Composition of the System

The hardware composition of pressure and displacement data acquisition system for press machine is shown in Fig.1. The component of the system mainly include pressure sensor, signal conditioning module, A/D converter ADAM4017, displacement sensor, counter, RS-485 convert to USB module and computer. The pressure sensor is used to detect the pressure signal and the conditioning module converts this signal to the appropriate voltage. ADAM4017 module converts this voltage into digital and sends them to the computer through RS-485. The displacement of press machine slide is detected by displacement sensor and transmitted to the counter. The counter, which connected to RS-485, sends the displacement signal to computer. Meanwhile, it also shows the displacement data and facilitates the operator to observe. RS-485 convert USB module converts RS-485 to USB and implements portable data acquisition. Computer creates the real time curve of pressure and displacement. The data of load and displacement during the part deformation can be stored into files. These data not only facilitate the offline analyze of processing procedures, but also provide a solid foundation for analyzing the status of part material and press machine.

---

<sup>+</sup> Corresponding author.  
E-mail address: kuxch@126.com.

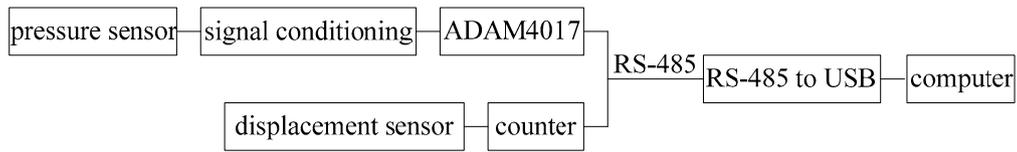


Figure 1. Hardware composition

## 2.1 Pressure sensor

It sealed with PTH type stainless steel. Its maximum pressure will reach 40Mpa and output 4~20mA current.

## 2.2 Signal conditioning

The isolated dual output signal distributor is used to convert 4~20mA current of pressure sensor into 1 ~ 5 V voltage signal.

## 2.3 ADAM4017 module

This module is developed by Advantech Co., Ltd. The ADAM Series is a set of intelligent sensor-to-computer interface modules containing built-in microprocessor. They are remotely controlled through a simple set of commands issued in ASCII format and transmitted in RS-485 protocol. They provide signal conditioning, isolation, ranging, A/D and D/A conversion, data comparison, and digital communication functions [1].

In this system, ADAM4017 is used to complete A/D conversion. ADAM4017 has 8 channels with optical isolation and its A/D conversion resolution is 16 bits. It can deals with voltage or current analog input signal and provides programmable input range for all channels. Because the 3000V isolation voltage is adopted to separate the input signal and internal circuit, the module will not be damaged when undergoing high voltage impact. The module has RS-485 communication function. Therefore, digital transmission is easily implementation.

## 2.4 Displacement sensor

The type of MPS-S-P rope displacement sensor is adopted in this system. This sensor has many advantages, such as integration characters of angle sensor and linear sensor, compactly structure, high precision and strong anti-interference ability. The sensor has maximum distance of 2000mm and generates A, B and Z pulse.

## 2.5 Intelligent counter

The type of XSN intelligent counter is used to count, display and transmit the pulse in this system. The counter has a high speed network communication interface. In addition, the special function of control transfer enables the monitor computer directly control the output of counter. The maximum count frequency reach to 10KHz and the read time for measurement data is less than 10ms.

## 3. Communication Protocol

The half-duplex mode is employed in this system to realize the communication between the monitor PC, displacement sensor and pressure sensor. The baud rate is 9600b/s. The communication way is: one start bit, 8 data bits, one stop bit and no parity. Two acquisition modules are given different address respectively. Monitor PC sends the command frame to acquisition modules. These modules judge if the command address is the same with itself when receive the command frame. If the address is same, the acquisition module collects data and returns the result to PC. Otherwise, the acquisition module continues keep in a listening state.

In this system, the communication protocols used by counter and ADAM4017 is quite different. So, PC needs select concrete communication protocol to realize data acquisition of displacement or pressure.

### 3.1 Communication command of counter

Its command frame format is as follows:

```
[start flag][address][content][constant][data][Checksum]
[end flag].
```

Where, the start flag is an initiation item. It indicates the beginning of the command frame. Each command must begin with start flag. There are six kinds of delimiter for the communication protocols, including #, \$, %, &, ' and ". The address follows the start flag. It designates the address of the counter with two characters. Content item used to specify the address of data or parameters. Constant item used to specify the command constant. Some commands need data, such as output command and setting parameters command. These data can be provided by the constant item. The checksum item used to check the command errors from the computer or answer errors from the counter. The checksum function adds additional two characters behind the string of command or answer. Checksum functions can be selected according to the real need. If the command contains the checksum, the counter will provide the answers that automatically add two characters. Checksum described with two ASCII code. The command checksum is equal to the total value of all the ASCII code in the command. If the number beyond the scope, the remainder data will be reserved. The checksum of the answer is equal to the sum of ASCII code of answers and the ASCII code of the counter address.

### 3.2 Communication command of ADAM module

The command format of the ADAM4000 series is as follows:

```
[start flag] [address] [command] [data] [Checksum] [CR].
```

All orders begin with a start flag. There are four kinds of start flag for the communication protocols, including \$, #, % and @. The address is two bit hexadecimal number. The checksum follows behind the command string. All order is ended by CR. For example, the command to read the data for ADAM4017 module is as follows: #02050x0d, where, the symbol # is the start flag, 02 is the address of ADAM4017, 05 is the fifth analog input channel. If the command correctly executing, the analog conversion value of fifth channel will return to computer.

## 4. Realization of Data Communication

### 4.1 Communication network of RS-485

The communication of the system is implemented with RS-485 network. It suitable for high speed and long distance transmit [2]. The master-slave protocol is adopted to manage the network in this system when computer communicate to ADAM and counter. The monitor computer controls the entire RS-485 operation and sends command. The ADAM and counter modules accept and respond to these commands. The input signal is isolated from the network using optical isolation circuit. Therefore, the modules are independent of each other. The reliability and security of the system greatly improved.

### 4.2 Serial port communication

Data acquisition and management in this system implemented through programming with Borland C++ Builder. BCB is a visual object-oriented application development tools for programmer and it provides a convenient development environment. There are three kinds of methods to realize serial communication [3][4]:

1) *The assembly language inserts into the C++ programs and directly operates the UART serial communication hardware.* This method has the highest real-time, but the programmer need to familiarize with hardware.

2) *Make use of the Windows API communication function.* This method has high flexibility and real-time, but the program is more complicated.

3) *Make use of VB high-performance ActiveX control MSComm.* This approach is simple and has a good real-time, but its program in binary mode is complicated.

No matter which method is used in the developing process of serial communication, its implementation steps are almost same. Overall consideration, the API communication functions is adopted to realize the acquisition of displacement and pressure in this system.

### 4.3 Data acquisition process

In this system, the host computer sends commands to modules according to the protocol. The modules carry out various operation when receive the commands, such as return the data to computer, parameters adjusting and data display. The computer using loop method to access ADAM4017 and XSN counter respectively and send order to them within a sampling cycle. Under the control of the host computer, the data exchanges between the computer and modules are accomplished and the data transmit in RS-485 network with the form of data frames. If some modules failure of disturbed and can not communicate with computer, the entire system may be abnormal. To avoid this situation, the command is sent to the port firstly, and then the watchdog process is started. It means the error has happened if the watchdog is overflow and the data still can not read. The failure module will be shown in the interface and notify the user to take measures in time to maintain the normal operation of the system. The flow chart of the data acquisition process is shown in Fig. 2

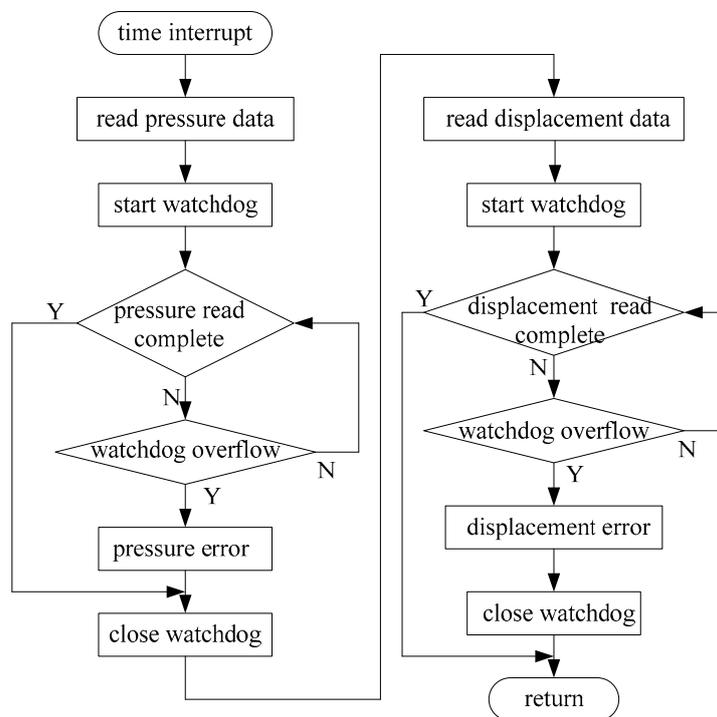


Figure 2. Data acquisition process

## 5. Program Design

### 5.1 Serial program in BCB

The serial communication implemented with Windows API functions in this system. API functions are provided by the operation system. Regardless of what kind of development tools, API functions will be used in the final stage. Because BCB has declared Win32 API, it can be directly used in BCB. Under the BCB environment, the major steps for serial program are as follows:

1) *Open the specified communication port with CreateFile( ) function.* This function indicates the application program has the possession right for this serial port and blocks other applications operate this port. The main parameters of this function are as follows:

```

HANDLE CreateFile (
LPCTSTR lpFileName, // file name
DWORD dwDesireAccess, //access mode
LPSECURITY_ATTRIBUTES lpSecurityAttributes, // share mode

```

```

LPSECURITY_ATTRIBUTES lpSecurityAttributes,
                        // pointer
DWORD dwCreationDistribution, //file open mode
DWORD dwFlagAndAttributes, // file attributes: 0
HANDLE hTemplateFile, //copy file: 0
)

```

If the operation is successful, the function will return a Handle value. The program will use the Handle value to operate this port.

2) *Set parameters and buffer.* Assign memory buffer for communication I/O queue using SetupComm( ) function. The communication parameters, such as baud rate, data bits, parity and stop bit, are configured by calling GetCommState( ) and SetupCommState( ) function.

GetCommState( ) function is used to read the serial port parameters, whose form as follows:

```

BOOL GetCommState(
HANDLE hFile, // handle of communications device
LPDCE lpDCB,
// address of device-control block structure
)

```

SetCommState( ) function is use to set parameters of serial port, whose form as follows:

```

BOOL SetCommState (
HANDLE hFile, // handle of communications device;
LPDCB lpDCB,
// address of device-control block structure,
)

```

3) *Realize read or write operation using ReadFile( ) function or WriteFile( ) function respectively.* For example, the format of WriteFile( ) is as follows:

```

BOOL WriteFile(
HANDLE hFile, // handle of file to write to
LPCVOID lpBuffer, // pointer to data to write to file
DWORD nNumberOfBytesToWrite,
// number of bytes to write
LPWORD lpNumberOfBytesWritten,
// pointer to number of bytes written
LPOVERLAPPED lpOverlapped,
// pointer to structure needed for overlapped I/O
)

```

The parameters of ReadFile( ) function are same with WriteFile( ) function.

## 5.2 Set the sampling cycle

During the process of data acquisition, the sampling cycle directly affect the sampling precision. Because Windows uses the multi-task mechanism, each process can not monopolize system resources. Therefore, how to ensure the accuracy of the sampling cycle is the key to achieving high precision data acquisition. Sampling cycle can be achieved with the Timer control of BCB. But the priority of Timer event is low and easily leads to the phenomenon of “lost seconds”. Meanwhile, its timing accuracy is only 55ms and cause the real time of dada acquisition is poor. In order to ensure the accuracy of sampling cycle, the system uses a more accurate multimedia timer. Multimedia timer can use its own single thread to call their separate callback function. It has a high priority. The minimum time accuracy is up to 1ms, so the multimedia timer can meet the requirement of minimal timing [5]. In the BCB environment, the usage steps of multimedia are as follows:

1) *define the timer parameter.* Firstly, add the header file mmsystem.h, then, add winmm.lib through the menu of Add to Project.

2) *Initialize and start the timer event using TimerSetEvent ( ) function.*

```

MMRESULT timeSetEvent( UINT uDelay, UINT uResolution, LPTIMECALLBACK lpTimeProc,
DWORD dwUser, UINT fuEvent );

```

Where, uDelay is used to set event delay; uResolution is resolution of the timer event; lpTimeProc is the address of a callback function; dwUser is the user-supplied callback data; fuEvent is the type of timer event.

3) *Statement and the preparation of the callback function TimeProc( ).* This function realizes the user function. The data acquisition of pressure and displacement in this system is achieved using this function.

```
Void CALLBACK TimeProc(UINT uID, UINT uMsg, DWORD dwUser, DWORD dw1, DWORD dw2);
```

Where, uID is the identifier of the timer event. This identifier was returned by the timeSetEvent function when the timer event was set up. uMsg is reserved by system. dwUser is the user instance data supplied to the dwUser parameter of timeSetEvent. dw1 and dw2 are not used.

4) *Delete the timer when the timer task is completed.*

```
timeKillEvent( Timer_ID ); //delete timer
```

The system creates a new thread after setting the multimedia timer. When the 50ms interval arrives, the interrupt occurred and the operation of reading and writing implemented. After the interrupt task is completed, the program come back and continues carry out the original mission. The function timeKillEvent( ) is employed to delete the timer and release the resources when exiting system.

### 5.3 Real-time curve

The system uses the chart component Tchart provided by BCB to realize the real-time curve of pressure and displacement. The method of using Tchart is as follows [6]:

1) *Initialization.*

```
Series1->Clear ( ) ;
```

```
Chart1->BottomAxis->Maximum=Max;
```

```
Chart1->BottomAxis->Minimum=Min;
```

2) *Draw curves in timer callback function.*

```
void CALLBACK TForm1::TimeProc(UINT uID, UINT uMsg, DWORD dwUser, DWORD dw1, DWORD dw2)
```

```
{
```

```
.....
```

```
Yvalue=PresData[i];
```

```
                //pressure data assigned to Y coordinate
```

```
i++;
```

```
Series1->Add(double Yvalue, AnsiString Xlable, Tcolor Color);           // X axis is Xlable, Y axis is
```

```
Yvalue;
```

```
Chart1->BottomAxis->Maximum= Chart1->BottomAxis->Maximum+1;
```

```
Chart1->BottomAxis->Minimum= Chart1->BottomAxis->Minimum+1;
```

```
..... // other relevant program;
```

```
}
```

Using the same method, the displacement curve and pressure-displacement curve can be drawn. These curves provide convenient for user to monitor the process of press working.

## 6. The System Interface

Fig.3 shows the interface of data acquisition. The interface is divided into two regions. The upper part mainly shows data curves, including displacement-time curve, pressure-time curve and displacement-pressure curve. The displacement-time curve and pressure-time curve are shown in the same window. These two curves respectively display the slider pressure and deformation of the workpiece produced when the press machine is working. The curve of displacement-pressure is displayed in another window. This curve directly shows the relationship between the pressure and deformation displacement. The zoom buttons placed on the curve right and used to enlarge or shrink the curve. The lower section contains some buttons that used to implement the data acquisition. These buttons include open, setup, clear, start and stop. The setup button is used to set the parameters of data acquisition, such as frequency, press machine selection, and so on. Start button realize the data acquisition according to the setting frequency. Clear button will clear the displacement data. This button's function is equivalent to set the relative zero for displacement sensor. Stop button will terminate the data acquisition and these data will be saved in text mode. User can use other software, such as Excel to open data files and process these data. Open button used to open previously saved data file and display these data with curves.

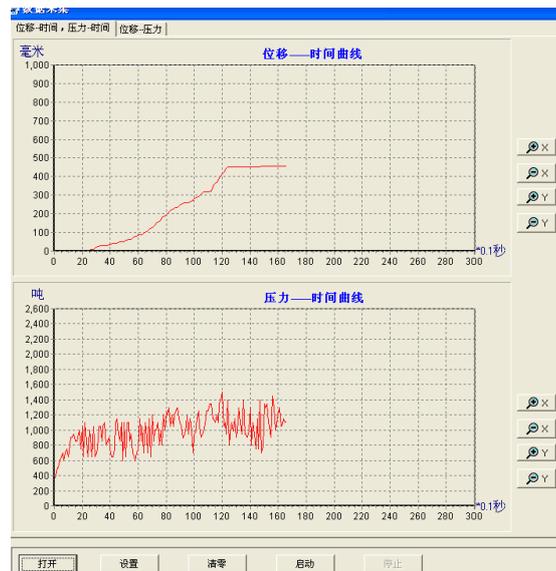


Figure 3. Interface of data acquisition

## 7. Conclusion

According to the actual production needs, this paper develops the data acquisition system based on RS-485. Through BCB programming, the data communication between PC and different data acquisition modules has been realized. At present, the system has been successfully applied to collect the data of pressure and displacement in the enterprise. The application result shows the system can record the data of pressure and displacement in the workpiece deformation process and meet the real-time and high precision measurement requirement.

## 8. Acknowledgment

The authors gratefully acknowledge the Key Project of Science and Technology Development of Henan Province (No. 082102210063), the Project of Nature Science Development of the education department of Henan Province (No. 2008A460006) and the Ph.D. Initial Foundation of Henan University of Science & Technology for financial support of this research work.

## 9. References

- [1] Advantech Co., Ltd, ADAM 4000 Series User's Manual, Taiwan, 2006.
- [2] SUN Peng, REN Wei-song, ZHANG Chi-bin, "Design of SBI Data Acquisition System Based on RS-485 Bus", Automation & Instrumentation, Sept.2009, pp.44-48.
- [3] WANG Juan, HUANG Yao-qun, "Realization of Serial Communication Program based on C++ Builder", Coal Technology, vol.26, Feb.2007, pp.24-26.
- [4] Fan Yizhi, Jiang Wenxian, Serial communication control for C++ Builder and RS232, Tsinghua university press, China, 2002.
- [5] Zhou Dongming, Su Xiaoguang, "IPC based Comprehensive Performance Detection System of Water Heater/ Dual-purpose Stove", Process Automation Instrumentation, vol.27, Mar.2006, pp.39-42.
- [6] Zhang Yao, Li Jihua, C++ Builder programming series, Metallurgical Industry Publishing House, China, 2002.