Sensors & Transducers, Vol. 172, Issue 6, June 2014, pp. 118-123

Sensors & Transducers

Design and Development of Coal Mining Equipment Inspection Instrument Application System

Chenguang Zhao, Boqiang Shi, * Shusen Liu, Mingchong Xu, Chentong Bian
School of mechanical engineering, University Of Science and Technology Beijing, 100083, China
* Tel.: 86-15652934357, fax: 86-010-62310302
* E-mail: liuss37@163.com

Received: 11 April 2014 /Accepted: 30 May 2014 /Published: 30 June 2014

Abstract: Based on the inspection instrument, the paper has offered an equipment inspection instrument regular repairing application system suited to the characteristics of coal enterprises. This system works stably in coal enterprises, effectively completing the measurement and management of equipment inspection, improving the reliability of the equipment, reducing the incidence of failure. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Coal mining equipment, Inspection management, Windows CE, Embedded system.

1. Introduction

Coal mining equipment inspection instrument is a portable device suits to the equipment management of the coal enterprises. Through the data interaction with the equipment inspection management system on the server-side, the equipment inspection instrument regular repairing can be realized through a corporate effort by the coal business equipment users and the equipment management departments. The equipment inspection instrument regular repairing system established in coal enterprises has brought an end to the traditional way of equipment managements, standardized the management of equipment inspection to achieve the procedures of management, the standardization of management business, the unification of the report file, the integrity of data and timely information processing [1]. The implementation of the equipment inspection management system in the coal mining enterprises can mobilize all personnel responsible for the equipment, which is of great significance to the safety and reliable operation of the equipment.

2. Overall System Design

The overall design of the equipment inspection instrument includes the hardware design and the software design. The hardware design should be analyzed according to the demand of the inspection instrument system and selects the appropriate hardware platform. The software design includes the selection and transplantation of the embedded operating system platform, as well as the realization of the features and applications of embedded systems based on the embedded hardware and embedded operating system.

2.1. The Selection of the Inspection Instrument Hardware Platforms

The inspection instrument hardware platform is the basis of the design and development of the system. Based on its functional requirement, the inspection instrument is mainly used in coal mine

which asks higher requirements of the explosion-proof type of the inspection instrument, it should be intrinsically safe instrument. Also the inspection instrument needs to complete the data collection, human-computer interaction, uploading and downloading functions and other functions. The hardware platform used includes the following components: a high-speed processor, memory, communication interface, touch screen LCD display, serial and power supply and other parts. So the Rainbow FeiDa EM3600C2 intelligent hand-held terminal is selected. Making S3C2440AL as it's core processor, articulated data acquisition as its peripherals, the handheld terminal features in explosion-proof coal safety certification, lower costs, stable in working and in line with the demand of the hardware platform. The inspection instrument hardware platform structure is shown in Fig. 1.

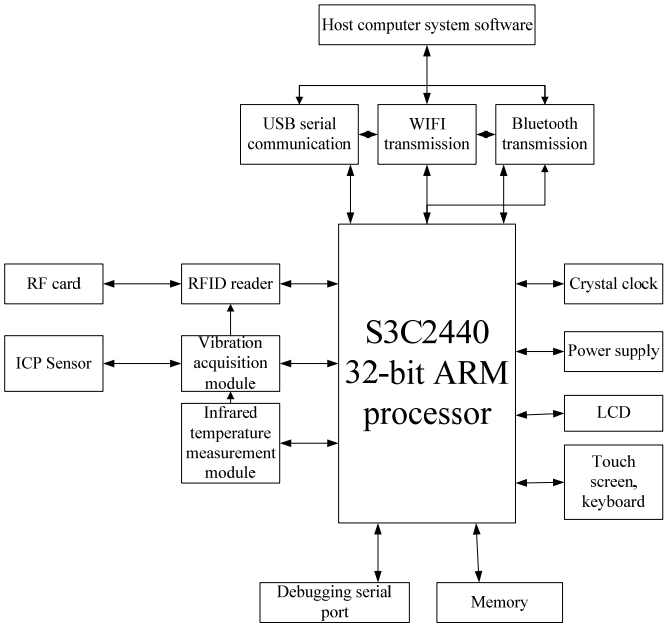

Fig. 1. The inspection instrument hardware platform structure.

2.2. Inspection Instrument System Functional Design

The inspection instrument system is developed based on the Windows CE operating system, with the system development platform using Visual Studio 2008 and the language development using Microsoft Visual C # language, internal database using SQLCE 3.5 embedded one, the overall database modeling analysis using PowerDesigner15 and using WINCE6.0 emulator to debug the system [2]. The human interaction is achieved mainly through a graphical interface. In line with the target for coal equipment management tailored according to the system requirements analysis, an inspection instrument application featuring in user-friendly, easy to operate is designed. The application system is divided into five functional modules: personnel login module, RFID regional card module, inspection instrument module, system communication module, auxiliary function modules. The overall architecture of the application software functional modules is given in Fig. 2.

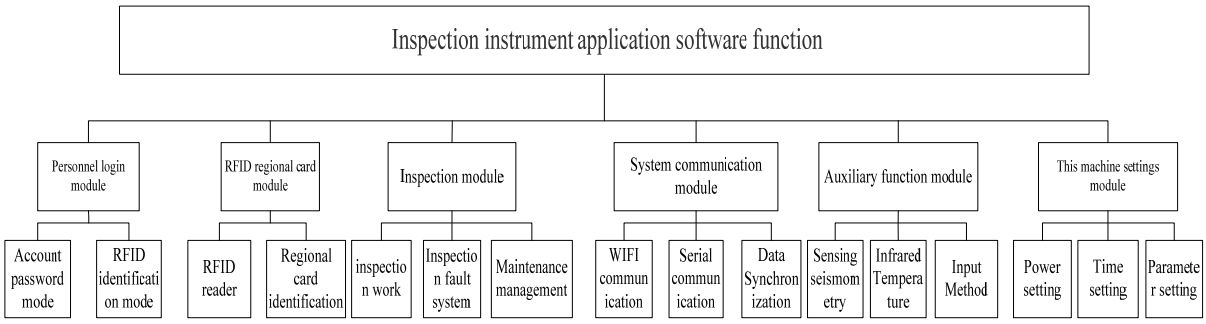

Fig. 2. Overall architecture of the application software functional modules.

2.3. System Interface Design

According to the function of system module design, 9 interface forms include the main interface form and the inspection interface form needs to be defined on the development platform, and the 9 interface forms load with each other to complete the inspection workflow. System interface diagram is given in Fig. 3.

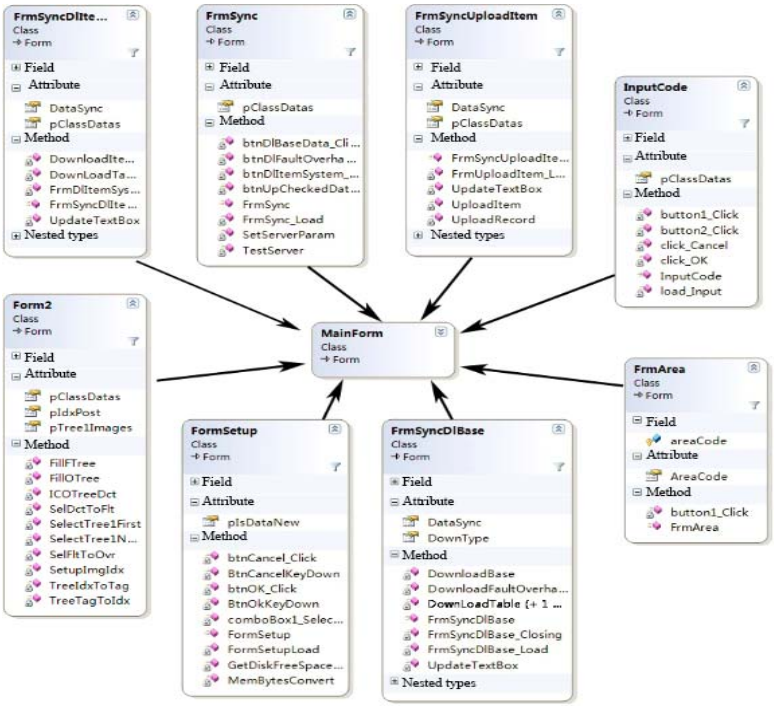

Fig. 3. System interface figure.

2.4. System User Login Interface Design

User login module uses the user name and password authentication login. Fig. 4 shows the login module UML figure, a detailed analysis of user login process. After entering a user name and password, the password into the database in an encrypted format, and read all of the user password information from the database, to verify whether there is a corresponding user password. If the verification is successful and filled with people testing data, if not then the authentication is unsuccessful and the need to re-enter user information.

According to UML figure flow design user login interface login module design follows the login screen, shown in Fig. 5.

After clicking the OK button, the system must work number and password to authenticate the user, while the employee login time into the database, make sure the button event code is as follows:

```
private void click_OK(object sender, EventArgs e) {
        if (gDataPerson.PersonLog(textBoxCode.Text.Trim(), textBox1.
                Text.Trim(), "", 0)) {
                label3.Text = gDataPerson.Czymc + ": Successful login"; }
        else  label3.Text = " Login failed "; }
```

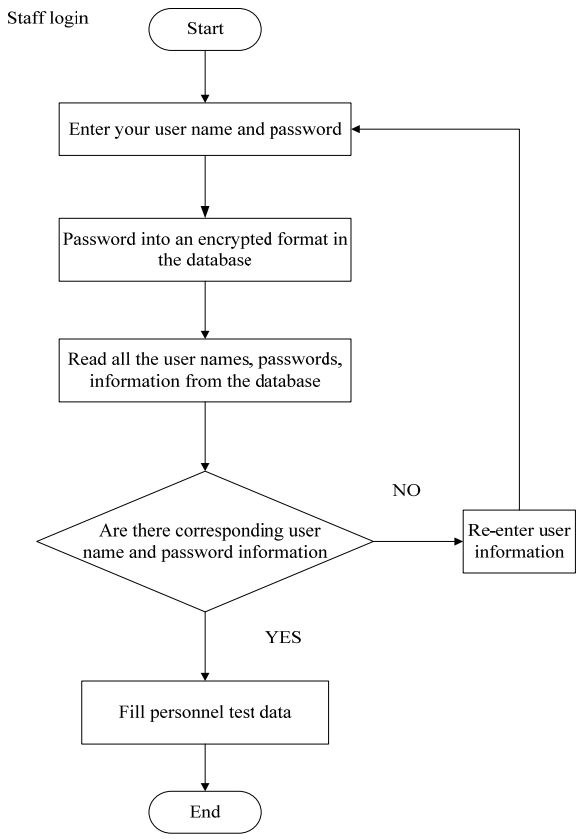

Fig. 4. Login module UML figure.

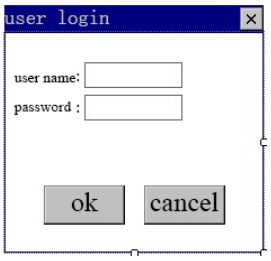

Fig. 5. User login interface figure.

PersonLog () method to verify the login information for the personnel function, this method is used to verify whether the login information for legitimate users, and in line with the user point inspection tasks all day job data added point inspection tree. Specific code to achieve this function is as follows:

```
public bool PersonLog(String Czybm, String Czmm, String RFID, int Mode) {
    String Sql = null;
    HelperDB hdb = new HelperDB();
    Czmm = Mmjm(Czmm); // Login password is encrypted
    if (Mode == 0) {
    // Mode 0 posts information by querying the database in line with the user name and password
    Sql = "select Gy_Czygl.Czymc,Gy_Czygl.Czybm,DEV_SI_Post.
                PostName,DEV_SI_Post.PostCode from DEV_SI_Post,
                DEV_SI_PersonPost,Gy_Czygl where DEV_SI_Post.
                PostCode=DEV_SI_PersonPost.PostCode and Gy_Czygl.
                Czybm=DEV_SI_PersonPost.Czybm and GY_Czygl.Czybm
                =\'" + Czybm + "\' and GY_Czygl.czmm=\'" + Czmm + "\'";}
    else if (Mode == 1) {
            // Mode 1 posts by querying the database in line with the RFID number of personnel information
    Sql = "select Gy_Czygl.Czymc,Gy_Czygl.Czybm,DEV_SI_Post.
                PostName,DEV_SI_Post.PostCode from DEV_SI_Post,
                DEV_SI_PersonPost,Gy_Czygl where DEV_SI_Post.
                PostCode=DEV_SI_PersonPost.PostCode and Gy_Czygl.
                Czybm=DEV_SI_PersonPost.Czybm and GY_Czygl.
                RFID=\'" +RFID + "\'"; }
    else {return false;}
    DataSet ds = hdb.GetDataSet(Sql, "Gy_Czygl");
            // Establish a data set and execute SQL statements
            int i = 0;
    Posts.Clear(); // Before adding the original post job information emptied information
    foreach (DataRow dr in ds.Tables[0].Rows) {
    // Cyclic query DataTable all rows of data
    Post post = new Post();
    if (i == 0) {
    this.Czybm = String.IsNullOrEmpty(dr["Czybm"].
                ToString()) ? "": dr["Czybm"].ToString().Trim();
    this.Czymc = String.IsNullOrEmpty(dr["Czymc"].
                ToString()) ? "": dr["Czymc"].ToString().Trim();
    this.Czmm = Czmm; }
    post.PostCode = String.IsNullOrEmpty(dr["PostCode"].ToString()) ?
                "" : dr["PostCode"].ToString().Trim();
    post.PostName = String.IsNullOrEmpty(dr["PostName"].
                ToString()) ? "" : dr["PostName"].ToString().
                Trim();
    Posts.Add(post);// The corresponding positions of the officer added Posts in
    i++; }
    this.IsLog = true;
    return true; }
```

2.5. System Home Page Design

The software page display of the inspection instrument application is divided into two levels. The first level of the home page shows the function of each module link, time, battery level and other information, the second level shows the contents page of each module. The home page design is divided into 16 regions and each module interface links are obtained through GotFocus. The converted icon is used to update the original icon after the focus and Set Full Screen function interface is used to achieve full screen display. Inspection instrument home page is shown in Fig. 6.

The home page code is as follows:

```
public static bool SetFullScreen(bool fullscreen, ref Rectangle rectOld) {
    if (fullscreen) {
      ShowWindow(Hwnd, SW_HIDE);
      Rectangle rectFull = Screen.PrimaryScreen.Bounds;
SystemParametersInfo(SPI_GETWORKAREA, 0, ref rectOld,
SPIF_UPDATEINIFILE);

SystemParametersInfo(SPI_SETWORKAREA, 0, ref rectFull,
SPIF_UPDATEINIFILE); }
      else{
        ShowWindow(Hwnd, SW_SHOW);
SystemParametersInfo(SPI_SETWORKAREA, 0, ref rectOld, SPIF_UPDATEINIFILE); }
      return true; }
```

(a) Initial state of the home page

(b) Home page focus state

Fig. 6. Inspection instrument home page.

2.6. Soft Keyboard Interface Design

In the software interface, automatic pop-up soft keyboard input if there is an edit box, and can automatically adjust the position of the soft keyboard when the edit box loses focus automatically close the soft keyboard. Software Input Panel (referred to as SIP) provides a way WINCE system data input. Interface design can be invoked when the soft keyboard SIP API function, where the main function is called SipShowIM, textBox_GotFocus when SipShow display software disk, textBox_LostFocus when SipHide hide the soft keyboard. The soft keyboard interface is shown in Fig.7. The following is the code that calls the soft keyboard:

```
[DllImport("coredll", EntryPoint = "SipShowIM")]
public bool SipShow()  {
    return SipShowIM(SIPStatus.SIPF_ON); }
public bool SipHide() {
    return SipShowIM(SIPStatus.SIPF_OFF); }
SIP softKey = new SIP();  // Generate soft keyboard object
private void textBox_GotFocus(object sender, EventArgs e) {
    softKey.SipShow();  // Display the soft keyboard }
private void textBox_LostFocus(object sender, EventArgs e) {
    softKey.SipHide();  // Hide soft keyboard }
```

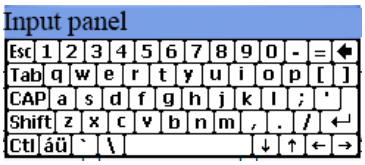

Fig. 7. Soft keyboard interface figure.

2.7. System Communication Interface Design

The inspection instrument system and server-side system commute mainly through the RDA data synchronously. Here the design of the system communication interface is mainly introduced. System communication can be connected via USB communication and WiFi communication. When using the USB communication, data communication can be achieved through inputting the computer name in the interface server box while inputting the server IP address and port number via WIFI communication.

The main consideration of the design of the system communication interface is whether abnormal circumstance occurs in the communication process and gives the user prompt in order to complete the normal connection. This paper designs the TestServer function to test SQL Server Compact Edition Client Agent errors that may be encountered when the RDA connect to SQL Server. When errors occur, the RDA connection is terminated and the users also get the error prompt. System communication interface is shown in Fig.8. TestServer function code is as follows:

```
public int TestServer() {
    int nRet = 0;
    SqlCeRemoteDataAccess rda = new SqlCeRemoteDataAccess();
    rda.InternetUrl = this.InternetUrl;
    rda.LocalConnectionString = Settings.AppSettings.ConnectionString;
    rda.InternetLogin = "";
    rda.InternetPassword = "";
    try { rda.SubmitSql("SELECT 0", this.OleDbConnectionString); }
    catch (SqlCeException ex) {
      switch (ex.NativeError) {
        case 28627: nRet = 1; break; // Unable to connect to the database
    case 28037: nRet = 2; break; // Unable to connect to the IIS proxy server
```

```
default:            nRet = ex.NativeError;  break; } }
    finally { rda.Dispose(); }
    return nRet; }
```

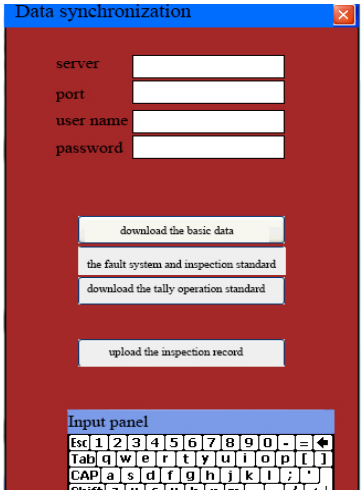

Fig. 8. System communication interface figure.

3. Conclusions

Through the research of the equipment management situation of coal enterprises, and in-depth research of the equipment inspection management in coal enterprise applications and embedded systems development, and detailed system requirements analysis of the inspection instrument, a handheld inspection instrument featuring in Chinese interface, easy to operate, fully functional, in line with the coal enterprise device management is designed and developed. The inspection instrument is an important part of the management systems of inspection. It can not only improve inspection staff efficiency, reduce the work strength of the inspection personnel and the management staff, but also increase the level of automated management of coal enterprises, effectively prevent over and less maintenance of the equipment, improve equipment reliability, reduce the incidence of failure.

References

[1]. Liu Weihua, Zhou Wei, Equipment inspection system management application and implementation, *Management and Fortune*, 2009, pp.85-88.

[2]. Lu Xiaofang, Windows CE-based power generation equipment inspection instrument software development, *North China Electric Power University*, Beijing, 2009.

[3]. Zhang Yingwei, Research of the mechanical equipment inspection instrument based on the embedded system, *Xi'an University of Science and Technology*, 2011.

[4]. Li Shican, Xu Qiqiang, Yang Maofu, Discussion on how to strengthen the coal mine electromechanical management, *Shandong Coal Science and Technology*, 04, 2010, pp. 237-238.

[5]. Mei Hong，Zhang Lu，Yang Fuqing. A component-based software configuration management model and its supporting system, *Journal of Computer Science and Technology*, 2002, 17, (4), pp. 432-441.

[6]. M. R. George, W. Ewong, Windows CE for a Reconfigurable System-on-a-Chip Processor, in *Proceedings of the International Conference on Field-Programmable Technology*, 2004. USA. pp. 201-207.

[7]. Okumura, S, Okino, N. A maintenance policy selection method for a critical single-unit item in each workstation composing a FMS with CBM optimisation, *International Journal of COMADEM*, 2003, 6, (2), pp. 3-9.

[8]. Jiang Weiwei, Research and system design of the model of the equipment inspection, *Anhui University of Science And Technology*, 2009.

[9]. Ye Junhua, Xu Xuemei, Huang Shuai, Guo Yuanwei, Xu Weiqin, Design of data acquisition system based on ARM and WinCE, *Computer Engineering and Science*, 2009, 31, (6), pp. 136-138.