Research on a monitoring terminal for a fibre grating sensing device based on Android

Xu Jiang [a], Lili Liu [c], Simeng Yu [b], Guohui Lyu [c,*], Xiaohan Zhan [c]

[a] *College of Information Science and Technology, Heilongjiang University, China*
[b] *Beijing University of Posts and Telecommunications, China*
[c] *Electronic Engineering College, Heilongjiang University, China*

Abstract

According to the actual needs of FBG sensing instruments in terms of intelligent terminals, software for FBG sensing monitoring systems is designed based on the current mainstream Android operating system, which runs on 3G mobile phones. The software is used to remotely access and manage a fibre optic sensing device. The use of the intelligent terminal software will enhance the level of safety monitoring instrument intelligence so that the management of the monitoring system will be more flexible, system maintenance will be more convenient, and the reliability and security of monitoring equipment can also improved; i.e., it provides good value for the optical fibre sensing application.
Copyright © 2014, Far Eastern Federal University, Kangnam University, Dalian University of Technology, Kokushikan University. Production and Hosting by Elsevier B.V. All rights reserved.

Keywords: Android; Intelligent terminal; Optical fibre sensing device; Data collection

Introduction

Optical fibre sensing techniques are an emerging technology that combines the optical, mechanical, electronics, materials and information processing. Optical fibre sensors possess the advantages of small volume, safe and durable operation, lightweight, immunity to electromagnetic interference, better insulation, no danger of deflagration, no power supply required at the sensor, remote measurement, and multi-point sensing. Optical fibre sensors are widely used in the petroleum, chemical, metallurgy, semiconductor, and aerospace industries and other fields. Currently, multi-functional fibre sensing devices are typically placed in the control room or a project site, making it inconvenient for engineering personnel to implement real-time operation, regular inspection, remote login and some other management activities; as a result, an intelligent terminal is required to solve this problem [1−4].

In this paper, based on a detailed analysis of the actual needs of FBG sensing instruments in terms of intelligent terminals, software was designed to operate the FBG sensing monitoring system of intelligent terminals on 3G mobile phones, which enables remote access and management of optical fibre

sensing devices. The software design is based on the current mainstream Android operating system. The software interfaces with a Fibre Bragg grating sensor monitoring system, which is used for data monitoring of optical fibre sensing devices. The design uses Socket communication technology, which utilises Wi-Fi wireless transmission to perform data communication and Android smart phone terminals to receive data; such data are then stored in a SQLite database, through which users can obtain historical data and plot the data to fully grasp the integrated operational performance of the optical fibre sensing devices. This paper also describes the design of an early warning function. When the monitoring data are beyond the intended scope, warning alarms will be sent to monitoring personnel. At that time, the warning function will inform the relevant worker to address the issue, thereby guaranteeing the safe operation of the equipment.

The 3G smart FBG sensing monitoring intelligent terminal, which is based on the Android operating system, was designed according to the actual needs of the project. This terminal will enhance the level of safety monitoring instrument intelligence by making the management of monitoring system more flexible; as a result, system maintenance will be more convenient and the reliability and security of monitoring equipment can also be improved; i.e., it has a good practical value.

Optical fibre sensing device data collection program

Optical fibre sensing systems are mainly composed of three parts, namely, the front sensors, the FBG demodulator, and the mobile terminal of an Android. An optical fibre sensing device monitoring system is schematically shown in Fig. 1.

The sensors are fixed at the monitoring site; these sensors monitor changes at regular time intervals. The raw data are transmitted to the optical fibre grating demodulator, and are then processed using a computer in the demodulator and stored.

This design uses a multi-channel demodulator, which uses advanced computing spectrum analysis technology to calculate the position of the spectral peak centre. The transmission of data is synchronised, as the demodulation devices regularly send data to the software. The standard transmission of data occurs 150 times per second, which corresponds to new data generated once every 6.66 ms. The software can submit a request for data to the demodulator once every
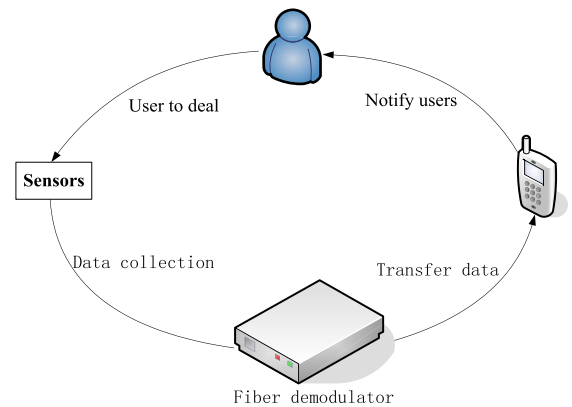

Fig. 1. Optical fibre sensing monitoring system.

4 ms; subsequently, the data are sent from the demodulator.

Within the Wi-Fi network coverage, the staff using the Android handset software can use the mobile terminal to send a request to the FBG demodulator. Once the fibre demodulator receives the request, the data can be transmitted through the socket channel. Subsequently, the data are received by the mobile terminal and then stored in the SQLite, which is the local database of the smart phone, for further analysis and display.

Intelligent monitoring terminal software design

Building the Android development environment

Android application development first involves establishing the development environment; in this study, this development occurred as follows:

(1) Configure JDK. First, download the JDK package, and then click to install. Next, configure the java environment variables;
(2) Configuration Eclipse. First, configure the Android SDK ADT dedicated to Eclipse, and then click on "Help-> Install New Software" and select "Add Site"; enter http://dl-ssl.google.com/android/eclipse/ in the "Location" dialog box and then Click on "Install" to install all of the Development Tools to find and install;
(3) Configure Android SDK. First, unzip the downloaded archive and the configuration environment variable. Next, open the Eclipse on the menu bar "Preference" option, click the Android tab, open "SDK Location:" and select the unzipped Android SDK directory; click on "Apply", and then set the Android SDK configuration;

(4) Create and run the Android emulator (AVD), and then select "Eclipse-Windows-Android AVD Manager" in Eclipse; next, click new, and then set the parameters that you want to individualise, including the JDK version, SD card size, screen size, etc. The SD card size is usually set to 512 MB [2].

The above is an overview of the built Android development environment. The operating Android emulator is shown in Fig. 2.

The design of the framework module

The client of the monitoring system of the optical fibre sensing device is primarily based on the MVC design pattern, namely, the model layer, the view layer and the control layer [3]. The MVC framework separates the interface display layer from the logic code and the data processing model layer. The advantages of this design are that the coupling is low, which will help maintain the system, and that the code reuse is high. The applications are independent of each module, and each application completes its own tasks, which is well suited for design and development based on Android. The MVC Framework of the applications is shown in Fig. 3.

(1) The view layer: Android provides a rich variety of UI components, which developers can conveniently use to develop user interfaces. Android
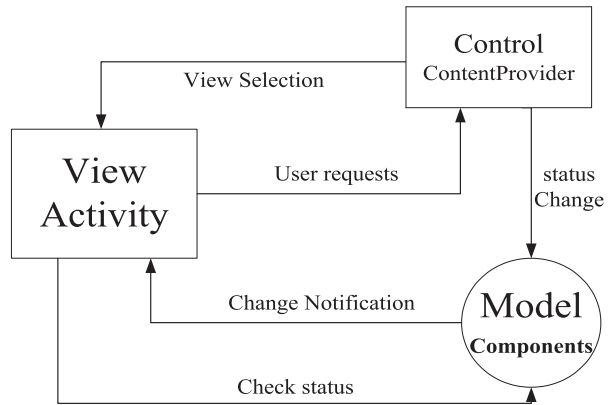

Fig. 3. The structure diagram of the MVC monitoring program.

contains most of the UI components in the Android.widget package and its sub-package. In Android, there are two methods to define the view layer components: one method is to Layout the XML attributes in the XML file; the other method is to call a method in a Java program. In this paper, the recommended Android XML layout file is used to define the interface.

(2) The model layer: operation of the database and the network are processed in the model layer. In this paper, we use ContentProvider( ) to share data and handle the details of the database and data processing, which enables an easy to maintain code.

(3) The control layer: for Android, the activity is generally from the controller, so the user interface and the control are integral. In the application, the
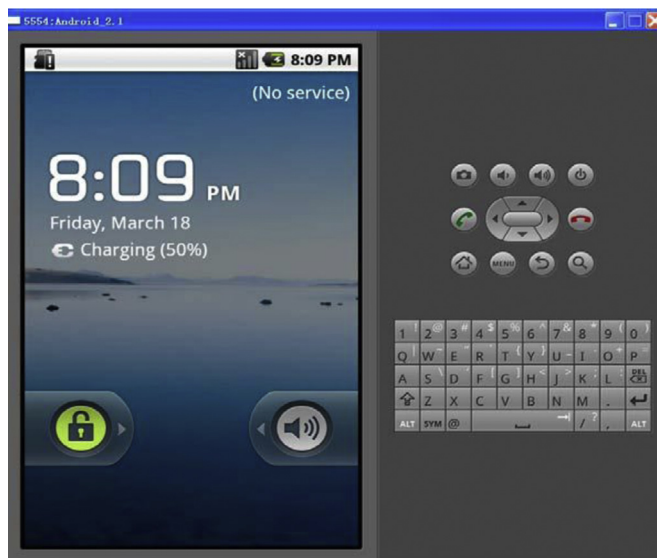

Fig. 2. Android emulator.

listener component and interface display both belong to the control layer.

The design of the function module

The monitoring system terminal functional block diagram of the system is shown in Fig. 4.

The functions of the intelligent terminal of an optical fibre sensing device primarily involve the receiving, analysis, storage and handling of the data. First, the Android phone smart terminal establishes a socket channel with the FBG demodulator; next, the client submits a request to the demodulator, and then the mobile terminal receives data from the demodulator.

Mobile terminals interact with the server through the established socket communication channel; part of the code is as follows:

Server establishes a connection to the remote server Socket:

Socket socket = new Socket ("192.168.1.13", 30000);

Client creates a ServerSocket, which is used to listen for connection requests of the client's socket:

ServerSocket ss = new ServerSocket (30000);

Maintaining the state of the client to monitor:

while (true) {…}

When the server receives the request from the client's Socket, upon receiving the request from the client Socket, it will generate a corresponding Socket:

Socket s = ss.accept ( );

After parsing the received data, obtain the wavelength data. The centre wavelength of the data is stored in the SQLite database. Part of the code to create the database is as follows:

Creating a database:

SQLiteDatabase db =

SQLiteDatabase . openOrCreateDatabase (

"/ src / com.fiber.db / fiber_database.db", null);

Creating the data table:

db.execSQL ("create table date_fiber (_id integer, Data_rev varchar (255),CreateDate varchar (50)), data_site varchar (10)");

Add and query data:

db.execSQL ("insert into date_fiber values (null,?,?,?)", New String [] {title, content});

Cursor cursor = db.rawQuery ("select * from date_fiber where createDate =?", Null);

Users can query the data and the historical curve by selecting the date. The data are retrieved using the SQL query statement. The intelligent terminal of the fibre optic sensing devices will also monitor the received data. When the received data exceed the warning threshold, the terminal will send warning signals to a preset phone number to notify the monitoring personnel to address the issue, thereby ensuring the safe operation of equipment.

The experimental results

The FBG monitoring software was tested on Android handsets after completion of the project code editing and debugging. The test process involves five parts, namely, login and registration, data collection, query historical data, query history curve, and warning function. The specific test procedures and the results are presented in the following.
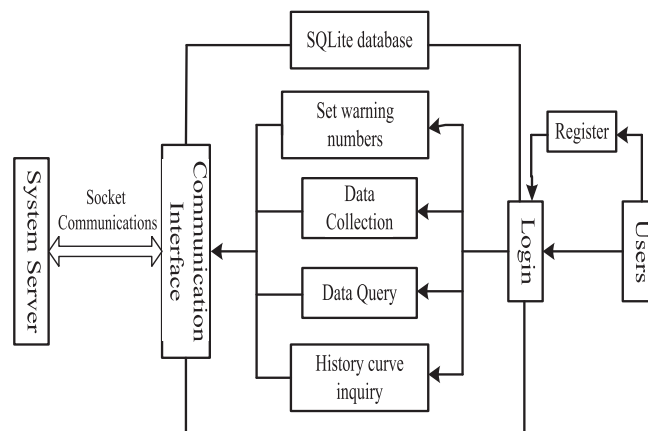

Fig. 4. The functional diagram of the monitoring system.
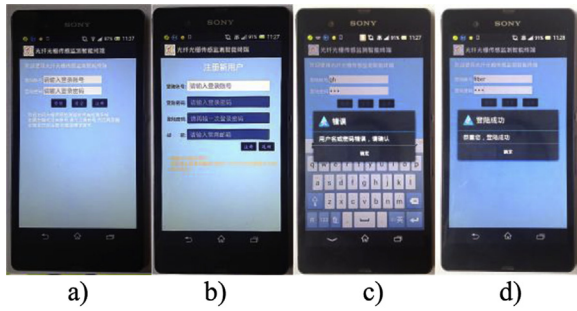
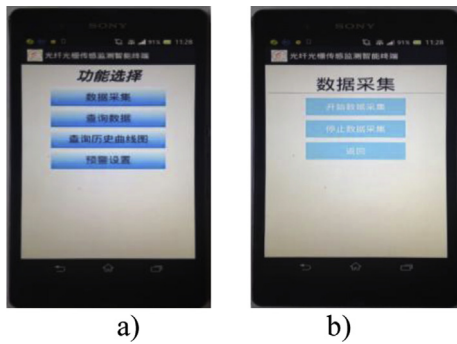Fig. 5. The test result of login and register.

Fig. 6. Sub-function interface.

First, open the login screen of the FBG monitoring intelligent terminal system. At this point, the new user must register for an account. If you already have an account, you can login directly. This application validates the username and password, with the test results shown in Fig. 5.

After the login successfully entered the sub-function interface, the sub-function interface is shown in Fig. 6a.

Click the button of data acquisition; next, entering the data acquisition interface, the data acquisition interface is shown in Fig. 6b.

Click the button for data acquisition to begin collecting data. If you want to stop the acquisition, click the button to stop data collection. By clicking on the Back button, the system returns to the sub-function interface.

Click the button labelled 'select the time' to select the date for which you want to query. This design verifies that for the date chosen, the date of selection cannot be empty. After you select the date, click OK, and then enter the data display interface; subsequently, you can see the average value along with the maximum and minimum values of each channel. The functional progress of the process is shown in Fig. 7.

Click the button labelled 'check the history' on the graph. Use the start and end dates to determine the range of the graph to be queried. Click the OK button to enter the history graph display. The operation process of inquiring for a historical curve is shown in Fig. 8.

Click on the button of the 'settings warning', and then set the phone numbers of the staff in the interface of setting warning phone. When the data are abnormal, the warning message will be sent to the relevant phones, thereby informing the staff to perform a processing operation.

Conclusions

The proper operation of the FBG sensing monitoring terminal applications was confirmed by experiment. In this paper, the application achieved the basic functions designed. The next step involves addressing the actual needs of the project to improve functionality. FBG sensing monitoring software can solve the

Fig. 7. Display data interface.

a)   b)   c)   d)

Fig. 8. The operational process of inquiring for a historical curve.

problem of real life and work that involve tedious manual operation, location and other aspects of the subject, thereby saving manpower and material resources; as a result, the software has broad prospects for development.

Acknowledgements

References

[1] P.M. Nellen, P. Anderegg, R. Broennimann, et al., Application of fiber optical and resistance strain gauges for long-term surveillance of civil engineering structures [C], in: Smart Structures and Materials' 97, International Society for Optics and Photonics, 1997, pp. 77–86.
[2] X.P. Liu, Y.B. Zhou, Design of an intelligent home system based on the android system [J], Appl. Mech. Mater. 443 (2014) 576–579.
[3] J. Vavru, M. Ujbanyai, Android Application Development: Complete Programming Guide [M], Grada Publishing a.s., 2014.
[4] H. Wang, Q. Chen, P. Wang, et al., Research and Implementation of the System Manager Based on Android Platform for Wireless Sensor Network [M], in: Advances in Wireless Sensor Networks, Springer, Berlin, Heidelberg, 2013, pp. 424–437.