

Beijing Geofanci Tech Co.,Ltd

SerialPort.jar API SPECIFICATION

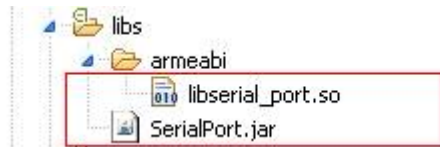
Xu bairu Diao xianhua
2014-4-2

目录

SerialPort.jar Specification	3
API Specification.....	3
1. SerialPort()	3
2. public void OpenSerial().....	3
3. public void OpenSerial ()	3
4. public int WriteSerialString().....	4
5. public int WriteSerialByte().....	4
6. public byte[] ReadSerial().....	4
7. public byte[] ReadSerial().....	4
8. public String ReadSerialSrting()	5
9. public String ReadSerialSrting()	5
10. public void CloseSerial()	5
11. public int getFd()	5
12. DeviceControl	6
12.1 DeviceControl().....	6
12.2 public void PowerOnDevice()	6
12.3 public void PowerOffDevice().....	6
12.4 public void TriggerOnDevice()	6
12.5 public void TriggerOffDevice().....	7
12.6 public void DeviceClose()	7

SerialPort.jar Specification

Instructions: this package has SerialPort class, there are four main functions:Open, Read, Write, Close. Encapsulate the different parameters and return types, a total of eight functions. Through jni calls libserial_port.so to achieve the serial port to read and write operations. Just put the libserial_port.so on libs/armeabi directory, you can call the functions packaged already due to import SerialPort package.



```
import java.util.TimerTask;
import android.serialport.SerialPort;
import android.app.Activity;
```

API Specification

1. SerialPort()

FUNC PROTOTYPE	SerialPort()
FUNC DESCRIPTION	Constructor of Class SerialPort.
RETURN VALUE	none

2. public void OpenSerial()

FUNC PROTOTYPE	public void OpenSerial(String dev, int baudrate) throws SecurityException, IOException		
FUNC DESCRIPTION	Open serialport device dev by baudrate, the default serial port parameters is Data bits 8, there is no check bit, Stop bit 1. Open succeeded, fd is the describe of the corresponding documents. Open failed, fd is negative.		
PARAM DESCRIPTION	dev	String	Describe the path of serial port device need be open
	baudrate	int	Specify the baud rate of serial port
RETURN VALUE	none		

3. public void OpenSerial ()

FUNC PROTOTYPE	public void OpenSerial (String device, int baudrate , int databit ,int stopbit,int crc) throws SecurityException, IOException		
FUNC DESCRIPTION	Open serialport device dev by baudrate, stopbit and crc. Open succeeded, fd is the describe of the corresponding documents. Open failed, fd is negative.		
PARAM	device	String	Describe the path of serial port device need be open

DESCRIPTION	baudrate	int	Specify the baud rate of serial port
	databit	int	Value is only 7 and 8
	stopbit	int	Specify the stop bit, value is only 1 or 2
	crc	int	Specify the check bit, 0: NONE, 1: EVEN parity, 2: ODD parity
RETURN VALUE	none		

4. public int WriteSerialString()

FUNC PROTOTYPE	public int WriteSerialString(int fd, String str, int len)		
FUNC DESCRIPTION	Write str length of len to the serial port fd, and returns the number of write successful.		
PARAM DESCRIPTION	fd	int	The file descriptor returned by using open to open device.
	str	String	The contents you want to write to serial port.
	len	int	Specify the length of the written string.
RETURN VALUE	int	The number of string written successful.	

5. public int WriteSerialByte()

FUNC PROTOTYPE	public int WriteSerialByte(int fd, byte[] data)		
FUNC DESCRIPTION	Write byte[] to the fd, returns the numbers of successful writing data.		
PARAM DESCRIPTION	fd	int	The file descriptor returned by using open to open device
	data	byte[]	The content you want to write to th serial port.
RETURN VALUE	int	The number of string written successful	

6. public byte[] ReadSerial()

FUNC PROTOTYPE	public byte[] ReadSerial(int fd, int len) throws UnsupportedOperationException		
FUNC DESCRIPTION	Read byte[] length of len from the serial port fd, the default delay is 50 milliseconds. Pay attention to the actual length of byte[], it may be less than len, or null represent there is none to read		
PARAM DESCRIPTION	fd	int	The file descriptor returned by using open to open device.
	len	int	maximum length need to read.
RETURN VALUE	byte[]	return byte[] read successful , it's length may be less than len. While null represent no data to read.	

7. public byte[] ReadSerial()

FUNC PROTOTYPE	public byte[] ReadSerial(int fd, int len, int delay) throws UnsupportedOperationException		
FUNC DESCRIPTION	Read byte[] length of len from the serial port fd, pay attention to the actual length of byte[], it may be less than len, or null represent there is none to		

	read		
PARAM DESCRIPTION	fd	int	The file descriptor returned by using open to open device.
	len	int	maximum length need to read.
	delay	int	The delay. Read data within the range delay, at the same time as far as possible to ensure data integrity.
RETURN VALUE	byte[]		return byte[] read successful , it's length may be less than len. While null represent no data to read.

8. public String ReadSerialSrting()

FUNC PROTOTYPE	public String ReadSerialSrting(int fd, int len) throws UnsupportedEncodingException		
FUNC DESCRIPTION	Read string length of len from the serial port fd, the default delay is 50 milliseconds. Pay attention to the actual length of string, it may be less than len, or null represent there is none to read.		
PARAM DESCRIPTION	fd	int	The file descriptor returned by using open to open device.
	len	int	maximum length need to read.
RETURN VALUE	String		

9. public String ReadSerialSrting()

FUNC PROTOTYPE	public String ReadSerialSrting(int fd, int len, int delay) throws UnsupportedEncodingException		
FUNC DESCRIPTION	Read string length of len from the serial port fd, pay attention to the actual length of string, it may be less than len, or null represent there is none to read.		
PARAM DESCRIPTION	fd	int	The file descriptor returned by using open to open device.
	len	int	maximum length need to read.
	delay	int	The delay. Read data within the range delay, at the same time as far as possible to ensure data integrity.
RETURN VALUE	String		

10. public void CloseSerial()

FUNC PROTOTYPE	public void CloseSerial(int fd)		
FUNC DESCRIPTION	Close the serial port fd.		
PARAM DESCRIPTION	none		
RETURN VALUE	none		

11. public int getFd()

FUNC PROTOTYPE	public int getFd()
FUNC DESCRIPTION	Return the file descriptor of operations.
PARAM DESCRIPTION	none
RETURN VALUE	int

12. DeviceControl

The class DeviceControl has several functions, through the way of writing device control file directly control equipment.

12.1 DeviceControl()

FUNC PROTOTYPE	DeviceControl(String path)		
FUNC DESCRIPTION	Constructor of Class DeviceControl. Open the device control file and set class data member CtrlFile correctly. Note: if failed to open device file, throws IOException exception.		
PARAM DESCRIPTION	path	String	Describes the equipment control file path of need to open
RETURN VALUE	none		

12.2 public void PowerOnDevice()

FUNC PROTOTYPE	public void PowerOnDevice() throw IOException		
FUNC DESCRIPTION	Call this function will power equipment and make it in working conditions. It will throw IOException exception when write device control file failed.		
PARAM DESCRIPTION	none		
RETURN VALUE	none		

12.3 public void PowerOffDevice()

FUNC PROTOTYPE	public void PowerOffDevice() throw IOException		
FUNC DESCRIPTION	Call this function will stop power to a device. It will throw IOException exception when write device control file failed.		
PARAM DESCRIPTION	none		
RETURN VALUE	none		

12.4 public void TriggerOnDevice()

FUNC PROTOTYPE	public void TriggerOffDevice() throw IOException		
FUNC DESCRIPTION	Call this function to make scan into scan state. It will throw IOException exception when write device control file failed.		

PARAM DESCRIPTION	none
RETURN VALUE	none

12.5 public void TriggerOffDevice()

FUNC PROTOTYPE	public void TriggerOnDevice() throw IOException
FUNC DESCRIPTION	Call this function to make scanner stop scanning. It will throw IOException exception when write device control file failed.
PARAM DESCRIPTION	none
RETURN VALUE	none

12.6 public void DeviceClose()

FUNC PROTOTYPE	public void DeviceClose() throw IOException
FUNC DESCRIPTION	Call this function to close the control file. It will throw IOException exception when write device control file failed.
PARAM DESCRIPTION	none
RETURN VALUE	none