



Barcode reader Nebula Software SDK USER GUIDE

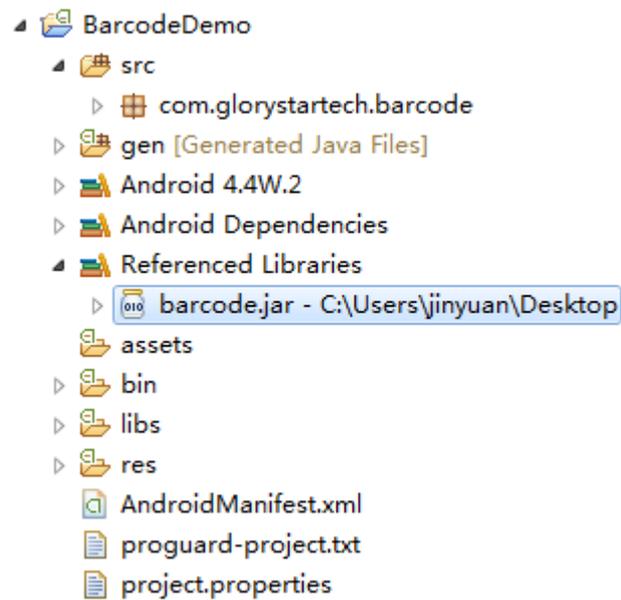


Introduction

The Barcode Scanner is a high performance 1D/2D scan engine that has outstanding scanning performance of its class. It provides flexibility in any kinds of applications, such as data collectors, kiosks and price checkers.

Barcode scanner must be adjusted to **USB COM Port Emulation mode**, run the APK, click **Open Device**, and it will display "successful open USB device!", that means that you have opened barcode settings, and can start scanning data.

1. Import the `barcode.jar` file into Referenced Libraries



2. Initialization in `MainActivity onCreate()`

```
// init
usbManager = (UsbManager) getSystemService(Context.USB_SERVICE);
btn_open = (Button) findViewById(R.id.btn_open);
btn_open.setOnClickListener(this);
edt_content = (EditText) findViewById(R.id.edt_content);
// Register BroadcastReceiver
this.registerReceiver(mBarcodeReceiver, new IntentFilter(
    ACTION_USB_PERMISSION));
```

3. In `btn_open`, Click the button to monitor the Barcode scanner

```
@Override
public void onClick(View v) {
    List<UsbSerialPort> serialPorts = getUsbSerialPorts();
    requestPermission(serialPorts);
}
```

4. Receive connection to USB equipment, find the barcode

```
// get USB Serialport
private List<UsbSerialPort> getUsbSerialPorts() {
    List<UsbSerialPort> mUsbSerialPorts = new ArrayList<UsbSerialPort>();
    List<UsbSerialDriver> mUsbSerialDrivers = UsbSerialProber
        .getDefaultProber().findAllDrivers(usbManager);
    for (UsbSerialDriver driver : mUsbSerialDrivers) {
        mUsbSerialPorts.addAll(driver.getPorts());
    }
    if (mUsbSerialPorts.isEmpty()) {
        Toast.makeText(getApplicationContext(),
            "Not connected to the usb device", Toast.LENGTH_SHORT)
            .show();
    }
    return mUsbSerialPorts;
}
```

5. After found the barcode, through the broadcast receiver, command to open.

```
// request usb device permission
private void requestPermission(List<UsbSerialPort> serialPorts) {
    PendingIntent pi = PendingIntent.getBroadcast(this, 0, new Intent(
        ACTION_USB_PERMISSION), 0);
    for (UsbSerialPort usbSerialPort : serialPorts) {
        this.usbSerialPort = usbSerialPort;
        UsbDevice usbDevice = usbSerialPort.getDriver().getDevice();
        usbManager.requestPermission(usbDevice, pi);
    }
}
```

6. Use **BroadcastReceiver** connect the **Barcode**

```
private BroadcastReceiver mBarcodeReceiver = new BroadcastReceiver() {

    @Override
    public void onReceive(Context context, Intent intent) {
        if (ACTION_USB_PERMISSION.equals(intent.getAction())) {
            synchronized (this) {
                UsbDevice usbDevice = (UsbDevice) intent
                    .getParcelableExtra(UsbManager.EXTRA_DEVICE);
                if (intent.getBooleanExtra(
                    UsbManager.EXTRA_PERMISSION_GRANTED, false)) {
                    if (usbDevice != null) {
                        openUsbDevice(usbSerialPort);
                    } else {
                        Toast.makeText(getApplicationContext(), "Error",
                            Toast.LENGTH_SHORT).show();
                    }
                }
            }
        }
    }
};
```

7. After successfully connected the Barcode scanner, start to read and scan data

```
// open usb device
private void openUsbDevice(UsbSerialPort usbSerialPort) {
    UsbDeviceConnection cn = usbManager.openDevice(usbSerialPort
        .getDriver().getDevice());
    if (cn == null) {
        return;
    }
    // Open the usb device to communicate
    try {
        usbSerialPort.open(cn);
        usbSerialPort.setParameters(115200, 8, UsbSerialPort.STOPBITS_1,
            UsbSerialPort.PARITY_NONE);
    } catch (Exception e) {
        e.printStackTrace();
    }
    // update SerialInputOutputManager
    stop(siom);
    open(siom);
}
```

8. First close the USB input and output management device

```
// Close the port stop IO read and write
private void stop(SerialInputOutputManager serialManager) {
    if (serialManager != null) {
        serialManager.stop();
        serialManager = null;
    }
}
```

9. Then, create USB input, output management device and input `usbSerialPort` and `mSerialListener`

```
// Open port for IO read-write
private void open(SerialInputOutputManager serialManager) {
    if (usbSerialPort != null) {
        siom = new SerialInputOutputManager(usbSerialPort, mSerialListener);
        mExecutorService.submit(siom);
        Toast.makeText(getApplicationContext(),
            "Successful open usb device", Toast.LENGTH_SHORT).show();
    }
}
```

10. mSerialListener monitor the Barcode scanner read data, and display the read data

```
SerialInputOutputManager.Listener mSerialListener = new SerialInputOutputManager.Listener() {  
  
    @Override  
    public void onRunError(Exception e) {  
        // Error  
    }  
  
    @Override  
    public void onNewData(final byte[] data) {  
        MainActivity.this.runOnUiThread(new Runnable() {  
            @Override  
            public void run() {  
                edt_content.setText(null);  
                edt_content.setText(HexDump.dumpHexString(data));  
            }  
        });  
    }  
};
```

11. When you quit the APK, must close the BroadcastReceiver and USB input output management device

```
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    unregisterReceiver(mBarcodeReceiver);  
    stop(siom);  
}
```

To find out more about our extensive range of Glorystar solutions, go to www.glorystargroup.com, or call the office nearest you. For more information, please contact us at sales@glorystargroup.com. Glorystar reserves the right to change or update, without notice, any information contained herein; to change, without notice, the design, construction, materials, processing or specifications of any products; and to discontinue or limit production or distribution of any products. This is only a reference document and does not apply to all hardware devices, Glorystar is not liable for any defaults /damages caused by this document or referenced from this document. Glorystar, the Glorystar logo, STAR CONTROL and STAROS are either trademarks or registered trademarks of Glorystar Group Limited. All other trademarks are the property of their respective owners.