

Použití RS232 na Android OS

Viktor Dlouhý, Ing. Jan Koprnický, Ph.D.

Abstrakt

Příspěvek se zabývá operačním systémem Android a jeho možnostmi v uplatnění na poli vestavěných systémů. Rozebírá možnosti zařízení vybaveným tímto systémem především v komunikaci s okolním světem. Stručně je rozebrána připravenost zařízení ovládat reálné digitální výstupy, komunikace přes TCP/IP. Největší důraz je věnován RS232 a to především USB RS232 čipu Profilic PL2303.

Úvod

Android je moderní operační systém vytvořený pro mobilní zařízení. Od doby prvního Androidu, který byl představen na nedotýkový mobilní telefon, uběhl už nějaký čas a mnoho verzí od té doby spatřilo světlo světa. Android byl implementován i na jiné zařízení jako televize, multimediální centra, hodinky či foťáky. Revoluční projekt Google Glass byl také představen na upravené verzi operačního systému Android. Tyto vlastovky přináší otázky ohledně možností tohoto mladého otevřeného systému. Je možné jednoduše tento systém použít jako vestavěný i na jiných zařízeních? Jaké jsou možnosti komunikace? Jak je Android připraven na RS232 - často používaný standard v průmyslu i jinde? Existují nějaké problémy v jeho implementaci? Na některé z těchto otázek odpovídá právě tento projekt.

Metodika

Většina zařízení s Androidem obsahuje USB, tato sériová sběrnice je podporována v API balíčkem `android.hardware.usb`. Tento balíček obsahuje několik tříd pro práci s USB zařízeními. Ne každý výrobce však zahrnuje do své verze systému veškeré ovladače, kterými je ve výchozí podobě Android vybaven. To je důležité nejprve prověřit. Ovladače zajišťuje Linuxový kernel.

Existuje v zásadě jen jedna možnost jak připojit RS232 na Android a to je pomocí RS232 USB konvertoru. Konvertor samozřejmě vyžaduje přítomnost USB hosta, většina tabletů a nových telefonů režim hosta umožňuje. Mezi RS232 konvertory se dá nelézt hned několik výrobců, ale v zásadě všichni používají čipy firem Profilic a FTDI. Tento projekt se zabývá čipem společnosti Profilic PL2303.

Výsledky a diskuze

Po prvním připojení konvertoru na zařízení s Androidem se většinou nestane vůbec nic, pakliže nějaká z nainstalovaných aplikací neobsahuje v sobě informaci, že umožňuje dané zařízení obsluhovat. Zařízení se identifikuje pomocí *vendor id*.

Tabulka 1. Rozložení pole bytů při inicializaci linky

Baud rate [4 bytes]	Stop bits [1 byte]	Parity [1 byte]	Data bits [1byte]
---------------------	--------------------	-----------------	-------------------

Při psaní vlastní aplikace pro komunikaci mezi jakýmkoliv zařízením podporující RS232 a například tabletem vybaveným Androidem, je nejdůležitější jedna věc – inicializace čipu. Nastavení potřebné rychlosti, parity a dalších vlastností linky je rozhodující pro funkčnost. Toto nastavení se provádí pomocí tak zvaného kontrol transferu. Taková metoda je ve třídě `UsbDeviceConnection` a umožňuje získávat a zapisovat nejrůznější nastavení čipu. Pro nastavení vlastností linky se používá

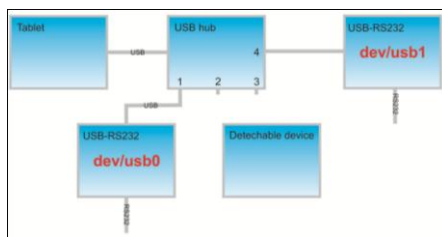
požadavek s konstantou 32 (resp. typ 33), který očekává hodnoty v poli bytů. Hodnoty se řadí v pořadí rychlost linky, počet stop bitů, parita a počet data bitů.

```
byte[] settings = new byte[7];  
//baudrate - 9600  
settings[0] = (byte) 128;  
settings[1] = 37;  
settings[2] = 0;  
settings[3] = 0;  
//stopBits - 1  
settings[4] = 0;  
//parity - even  
settings[5] = 2;  
//databits - 8  
settings[6] = 8;  
this.connection.controlTransfer(33, 32, 0, 0, settings, 7, 100);  
//turn off flow control  
this.connection.controlTransfer(33, 34, 0, 0, null, 0, 100);
```

Kód 1. Příklad zaslání kontrol transferu na PL2303

Po nastavení čipu je už možné používat klasické metody z USB balíčku Android API pro přenos dat jako `bulkTransfer()` a jiné. Ty už nepředstavují obtíže. Je vhodné ještě upozornit na ošetření *povolení*, které je nutné přidělit každému zařízení připojenému přes USB (pokud to není USB mass storage) k Androidu. Toto povolení je možné věnovat aplikaci buď dočasně, nebo natrvalo. Vždycky je však vyžadována účast uživatele a aplikace ve výchozí formě nemůže mít stálé *povolení* k zacházení s USB zařízením.

Pokud je třeba použít více RS232 konvertorů, není možné exaktně odlišit jednotlivá zařízení od sebe, jelikož mají stejné *vendor id*. To je částečně možné řešit pomocí portu.



Obrázek 1. Identifikace pomocí portu

Chybou operačního systému je listování připojených USB zařízení. Třída `UsbManager` umožňuje procházet jednotlivě z nich. Problém však nastává, že Android nedokáže registrovat událost odpojení USB a však událost připojení je funkční. Toto je již nahlášená chyba, která se projevuje v tom, že událost odpojení nefunguje. Proto zůstávají v seznamu i již dávno odpojené USB.

Závěr

V práci se úspěšně podařilo zmapovat možnost použití RS232 na operačním systému Android. Během práce se objevily nejrůznější problémy spojené především se špatnou dokumentací USB balíčku v Android API, ale i jiné zapříčiněné chybami systému.

Poděkování

Děkuji panu Ing. Janu Koprnickému, Ph.D., že mi umožnil tuto práci realizovat.

Reference

- [1] USB Host – API Overview. Developers Android. [Online] [Citováno: May 5, 2013.] <http://developer.android.com/guide/topics/connectivity/usb/host.html>.
- [2] Serial and UART Tutorial. FreeBSD. [Online] [Citováno: May 1, 2013.] <http://www.freebsd.org/doc/en/articles/serial-uart/>.