

Připojení modulů IQRF k platformě Android za použití USB nebo SPI

Bc. Josef Jebavý, <http://xeres.cz>

27. října 2011



Obsah

1	Operační systém Android	2
2	Možnosti řešení	2
2.1	USB	2
2.1.1	USB Host	2
2.1.2	Accessory	2
2.1.3	Vlastní řešení USB	3
2.2	SPI	3
3	Volba řešení	3
4	Hardwarové nároky vývojového pracoviště	4
5	Význam použitých zkratk a termínů	4

1 Operační systém Android

Android je OS určený pro mobilní telefony a aktuálně už i tablety. Tento operační systém je určen pro hardware s procesory Arm verze 5 a novější. Je založen na Linuxu, čili kernelu linuxových systémů. Tento kernel je značně upraven a tak už s Linuxem není 100% kompatibilní. Android využívá i některé GNU komponenty, ale s linuxovými distribucemi má společného málo. Android je psán převážně v jazyce C/C++ a aplikace pro něj (dále nazývané jako nativní) se píšou v jazyce, který je podobný Javě. Android je dobře použitelná platforma pro spotřební uživatele i pro nemalou skupinu vývojářů a proto si myslím, že se stala zajímavou pro firmy ať už pro použití samostatně nebo v našem případě s připojenými moduly IQRF.

2 Možnosti řešení

Jak propojit IQRF s Androidem a provést to zároveň rychle? Nabízí se dvě základní možnosti připojení IQRF modulů pomocí USB a nebo pomocí SPI.

2.1 USB

Dle oficiální dokumentace Android umí dva druhy připojení externích zařízení pomocí USB rozhraní: **USB Host** a **Accessory**.

2.1.1 USB Host

Pro podporu této komunikace je potřeba mít Android verze 3.1 (API level 12) a vyšší. V tomto módu se zařízení chová jako standardní počítač (host), což není běžná věc u mobilů a ani to běžně mobilní telefony neumějí. Zásadní věc je, aby hardware podporoval USB host u mobilních zařízení se obvykle jedná o USB OTG. Takto by se pak dalo připojit USB IQRF zařízení v módu CDC nebo mass-storage, pro které už i Android obsahuje podporu.

2.1.2 Accessory

Podporu této komunikace obsahuje Android verze 3.1 a byla backportována i na Android 2.3.4 (API level 10). Toto je zvláštní typ komunikace, kdy se telefon chová jako zařízení (device) a připojená periférie jako počítač. Není tak na straně Androidu potřeba ovladač ani nepotřebujeme aby hardware uměl USB host. Řešení problémů zprovoznění se tak částečně přenáší na připojený hardware. Existují různá vzorová řešení převážně s použitím procesorů Atmel, většinou založené na platformě Arduino. Ale protože moduly IQRF používají procesory od firmy Microchip, tak zde uvádím odkaz na řešení s procesorem a vzorový kit od téže firmy: [3-step Approach to Develop Android Accessories](#). Předpokládám, že pro toho kdo má zkušenosti s mikroprocesory firmy Microchip, by s tímto řešením neměl problém.

2.1.3 Vlastní řešení USB

Jelikož Android vychází z Linuxu dá se předpokládat, že by také Android mohl mít sice navenek skrytou, ale uvnitř existující podporu USB jako je v Linuxu. Při podrobném průzkumu internetu se dá narazit na zmínky, že i na starších verzích Androidu bylo zprovozněno USB rozhraní. Podrobnější popis však většinou chybí. Z průzkumu zdrojových kódů Androidu, jsem zjistil, že kernel Androidu obsahuje různé zdrojové kódy pro USB a vypadá to i na podporu USB host. Takže by bylo možno portovat ovladač konkrétního externího USB zařízení z Linuxu na Android. Konečnou prací by pak bylo vytvoření vrstvy, která by poskytovala nativním aplikacím API pro komunikaci s IQRF modulem. Základní podmínkou je zde opět podpora USB host a aby byl k dispozici ovladač USB řadiče a ovladač pro USB zařízení IQRF modulu.

O připojování USB zařízení k mobilním telefonům se dá dočíst i v mém článku „Speciální redukce USB se zdrojem“ v časopise Praktická elektronika - Amatérské Radio 2011/7 strana 20.

2.2 SPI

Zatímco USB je rozhraní určené pro uživatele a pro připojování periférií, SPI je rozhraní ryze hardwarové. Jeho výhodou je to, že je to základní rozhraní modulů IQRF, a tak není potřeba dalšího HW na straně modulu případně ani telefonu či tabletu. Mobilní telefony a tablety běžně obsahují SPI a používají ho právě pro připojení hardwaru. Základní podpora SPI, by tak v Androidu měla být, to potvrzuje i průzkum zdrojových kódů. Pokud by byla v zařízení volná linka SPI, realizace by se skládala z naprogramování systémové vrstvy, která by propojovala IQRF modul s vnějším světem. Konkrétně by vytvářela vnější API pro nativní aplikace a spodní vrstva by byla napojena na SPI vrstvu systému nebo přímo na vrstvu v kernelu.

Přesto, že již mám telefon již k modulům IQRF připojen, tak toto řešení již dlouho plánuji provést a to nejen na mobilním telefonu FreeRunneru ale i tabletu TouchBook.

3 Volba řešení

Nejrychlejším řešením se tak stává volba USB host, což by vyžadovalo vytvořit knihovnu pro komunikaci se zařízením v módu CDC nebo mass-storage. Podmínkou je však mít novou verzi Androidu a aby zařízení umělo USB host. Pokud zařízení neumí USB host, tak je dobrou volbou řešení úprava USB IQRF modulu, tak aby se choval jako Accessory. Obě tato řešení by byla implementována jako nativní Java knihovny s užitím Android Java API, a tak by nevyžadovaly žádný zásah do systému a proto by ho pak bylo možné použít i na libovolném zařízení.

Zařízení by mělo mít vždy aktuální systém obzvlášť v době uvedení na trh, pokud však obsahuje starý Android připadá v úvahu pouze SPI nebo vlastní zprovoznění USB. SPI je zajímavou volbou. Z pohledu hardware-vývojáře se jedná o čisté řešení, ale vyžaduje velký a pracný zásah do systému. Možnost zprovoznění USB podpory do starších verzí Androidu než 3.1 a 2.3.4 je realizovatelné a mělo by být předmětem

dalšího výzkumu a testů. Pokud je USB subsystém na Androidu málo odlišný od původního Linuxu, je vysoce pravděpodobné, že zprovoznění IQRF modulů by bylo podobné jako na čistých linuxových platformách např: IQRF na OpenMoko Neo Free-runneru . Obě tato řešení by byla implementována jako C/C++ ovladač, knihovna a API pro nativní aplikace. Nevýhodou těchto řešení je i pomalejší vývoj v jazyce C/C++ a nutnost zabývat se jak vnějším tak vnitřním API. Vzhledem k tomu, že s SPI mám málo zkušeností a že IQRF pomocí USB se už podařilo propojit s Linuxem, čili existuje aspoň jeden ovladač, pak bych osobně z hlediska rychlejšího zprovoznění volil USB.

I když čistší řešení se mě zdá použití SPI, obzvláště v případě telefonů s GNU/Linuxovým systémem. Které kromě firmy OpenMoko nabízí i například firmy Nokia a bývalí Palm. Úspora součástek, je také výhodná. A to z pohledu ceny, místa a minimalizaci problémů.

4 Hardwarové nároky vývojového pracoviště

Vývoj v oblasti mobilních počítačů, tak jako je Android, je náročný i z hlediska výkonu, proto je potřeba mít pro vývoj dostatečně výkonný hardware obzvlášť pokud se nejedná jen o aplikace, ale i o úpravy systému. Minimální hardwarové zázemí pak obsahuje počítač s výkonným CPU o více jádrech, velké množství RAM a pro zajištění dobré diskové propustnosti disky v RAIDu 0 nebo kvalitní SSD. Vhodné je mít vyčleněn stroj jen pro kompilaci. Samozřejmostí je kvalitní zařízení s Androidem, který je od výrobce odladěný a dokumentace k němu. Vše pod operačním systémem Linux.

5 Význam použitých zkratk a termínů

API application programming interface

ARM je architektura procesorů, vyvinutá v Británii firmou ARM Limited

CDC communications device class

CPU central processing unit

GNU GNU's Not Unix!

IQRF platforma pro jednoduchou bezdrátovou komunikaci

mass-storage protokol umožňující na USB přenášet soubory (flash paměť)

OS operační systém

RAID Redundant Array of Independent Disks

RAM Random Access Memory - paměť s libovolným ("náhodným") přístupem

SPI Serial Peripheral Interface Bus

SSD solid-state drive

USB Universal Serial Bus

USB OTG USB On-The-Go, zařízení umí mód host i device