Vysoké učení technické v Brně
Fakulta informačních technologií

Softwarová simulace robota řízeného FPGA

15. září 2011 Lucie Matušová

Obsah

1 Úvod

Tento text popisuje testovací plaformu výzkumu fault tolerant na UPSY, FIT. Jde o softwarovou simulaci robota, jehož chování je řízeno FPGA. Komunikace FPGA a počítače probíhá přes sériové rozhraní FITkitů.

V této zprávě najdete seznámení s projektem Player/Stage a jeho instalací, popis návrhu a implementace řízení robota včetně komunikačního rozhraní. V závěru se nachází také pár poznámek týkajících se řešení problémů, na které může uživatel při práci s platformou narazit.

2 Player project

Většina informací byla čerpána z [2], kde jsou k dispozici návody k instalaci, odkazy na soubory ke stažení, Wiki atd. Pro seznámení s funkčností a možnostmi Player/Stage je výborný manuál v angličtině od Jenny Owen [6]. Obsahuje také příklad robota, který byl použit jako řídicí základ pro tuto testovací platformu.

Projekt Player zahrnuje *Player* server, simulační prostředí *Stage* (2D) a *Gazebo* (3D). Tento software je vydán pod GNU General Public License.

2.1 Player robot device interface

Player je server pro řízení robotů. Poskytuje jednoduché rozhraní senzorům robota a realizuje spojení přes IP síť. V našem případě spolupracuje se Stage na simulaci *světa* a automaticky zapíná Stage plugin.

Spuštění: `player bigbob.cfg`

2.2 Stage multiple robot simulator

Stage umí simulovat roboty, kteří se pohybují a snímají 2D bitmapové prostředí. Je to klient, který komunikuje se serverem, Playerem, přes TCP socket.

Roboti mohou mít na sobě umístěny různé senzory jako např. sonar, skenovací laser nebo kameru umožňující detekci "kapek"(tzv. blobs). Tato hardwarová zařízení jsou připojena k serveru jako klienti přes tzv. *proxy*. Kód pro řízení robota se musí připojit na server a až potom může k těmto proxies přistupovat. Proxies, které má robot připojeny, musí být specifikovány v `.cfg` souboru, který je mapuje na zařízení v Playeru, jinak k nim nelze přistupovat.

Roboti se pohybují ve světech definovaných soubory `.world`. V těchto souborech se nachází také popis jejich senzorů.

Zdrojové soubory Stage spolu s ukázkovými `.world` a `.cfg` soubory a aktualizované informace ohledně instalace jsou ke stažení na githubu [4].

2.2.1 Proxies

Pouze pár poznámek k proxies zařízení použitých v našem robotovi. Podrobnější popis a další druhy proxies naleznete v [6].

Sonar proxy se dá použít k získání vzdálenosti od překážky. Vrací vzdálenost v metrech.
Laser proxy "skenuje"zadaný úhel v prostoru.
Blobfinder proxy analyzuje obraz z kamery a hledá místa s výskytem dané barvy. Vrací pole struktur, kde jsou uložena tzv. blob data.

3 Instalace

Funkční instalace Player/Stage běžela na Ubuntu Karmic 9.10 a s trochou štěstí i na novějším Ubuntu Natty 11.04. Instalaci lze provést dvěma způsoby: použít neoficiální *balíčky* nebo si programy svépomoci *zkompilovat*.

3.1 Balíky

Neoficiální balíček, který obsahuje, mimo jiné, Player a Stage, se nachází na [3]. Naleznete tam také instrukce k instalaci software z PPA. Po přidání PPA do systému stačí pro instalaci Player a Stage napsat: `apt-get install robot-player stage`.

Tento způsob instalace fungoval na Ubuntu 11.04.

3.2 Kompilace ze zdrojových kódů

Při kompilaci je pro Player a Stage zapotřebí několik knihoven a nástrojů, což vede k problémům napříč různými verzemi.

S následující konfigurací byly úspěšně nainstalovány Player a Stage v Ubuntu 9.10.

Stage (3.2.2) [5]

- cmake
- ltdl (libtool)
- OpenGL
- pkg-config
- libfltk >=1.1
- libpng
- jpeglib

Player (3.0.2)

- boost 1.38 (thread, signals) [1]

[1] Pozn. Problémy spojené s vyšší verzí knihovny boost (1.42) "vyřešila" instalace z balíčků.

4 Analýza a implementace

4.1 Požadavky

Úkolem je propojit softwarovou simulaci robota a FPGA. Simulace zasílá senzorická data do vývojové desky a přijímá od ní signály, kterými je řízena. Existují tedy čtyři části - zaslání senzorických dat do desky s FPGA, řídicí logika v FPGA, přijetí řídicích signálů v simulaci a provedení akce.

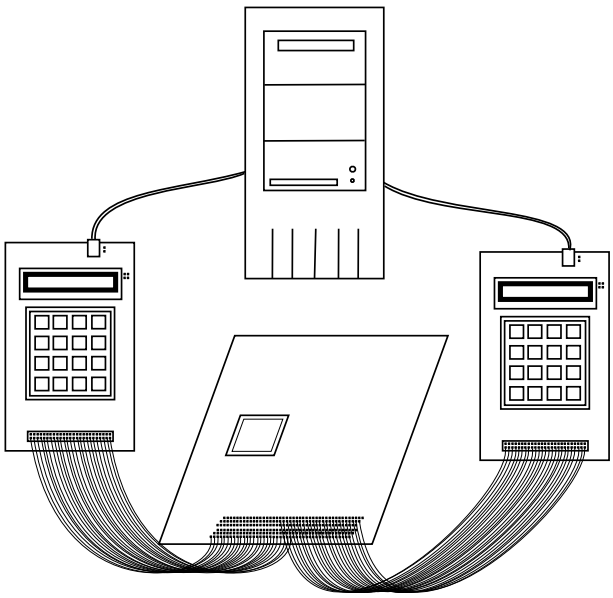
4.2 Prostředky

Pro komunikaci desky s počítačem máme k dispozici FITkity (1.2) [1]. Jako vývojovou desku jsme zvolili ML506 s FPGA Virtex-5.

K naprogramování FITkitů je zapotřebí mít zprovozněnou aplikaci *QDevkit* v nejnovější verzi, která umožňuje vzdálený překlad. Pokud zvolíte starší verzi, musíte ještě nainstalovat xilinxovské nástroje pro překlad. Ty jsou nutné také pro překlad řídicí jednotky robota. Pro naprogramování FPGA použijeme nástroj *Impact*.

4.3 Zapojení

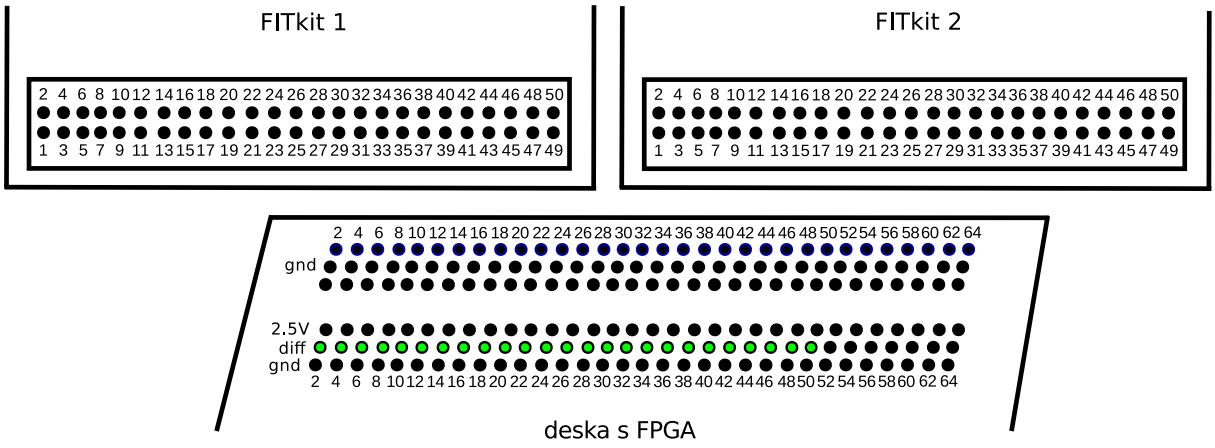
Data z počítače jdou do sériového portu prvního FITkitu, který je pošle po SPI do pinheaderu. Co se týče počtu senzorů robota, je při jeho vytváření třeba vzít v potaz poměrně nízký počet využitelných pinů FITkitu (40). FITkit je spojen s vývojovou deskou. Senzorická data jsou zpracována v FPGA a do druhého FITkitu jsou zaslány řídicí signály. FITkit je pošle zpět do počítače, kde je na jejich základě provedena akce (např. posun robota).



Obrázek 1: Zapojení

4.4 Komunikace

Ucelené informace o komunikačním systému FITkitu se nacházejí v [1], sekce *firmware*. V této části zběžně popíšeme způsob, jakým putují data ze simulace do FPGA a zpět do počítače.



Obrázek 2: Piny na FITkitu a desce

FITkit 1	**Deska**	**Deska**	**FITkit 2**
11, 13, …, 23, 25	2, 4, …, 14, 16	18, 20, …, 30, 32 *(diff)*	11, 13, …, 23, 25
12, 14, …, 24, 26	18, 20, …, 30, 32	34, 36, …, 46, 48 *(diff)*	12, 14, …, 24, 26
27, 29, …, 39, 41	34, 36, …, 46, 48	50 *(diff)*	27
28, 30, …, 40, 42	50, 52, …, 62, 64		
43, 44, …, 49, 50	2, 4, …, 14, 16 *(diff)*		

Tabulka 1: Spojení pinů

4.4.1 PC -> FITkit

Program v počítači otevře sériové spojení s FITkitem. Baud rate FITkitu je 460800 Bd a nastavuje se následovně:

```
cfsetospeed(&tty_attributes,B460800);  cfsetispeed(&tty_attributes,B460800);
```

Naměřená senzorická data se přes sériový port pošlou do FITkitu. Mikrokontroler FITkitu zachytí data ze sériového portu pomocí funkce `decode_user_cmd` pro dekódování uživatelských příkazů a jejich vykonávání. Vyparsuje se typ senzoru a jeho aktuální hodnota. Funkcí `FPGA_SPI_RW_A8_DN` se hodnota pošle po SPI na adresu danou typem senzoru. Tato adresa je ve vhdl souboru namapována na pinheader. Např. na piny 11, 13, …, 25 je vyvedena hodnota prvního senzoru.

4.4.2 FITkit -> vývojová deska a naopak

V této části je zapotřebí pouze správně namapovat signály na pinheader FITkitu a v `.ucf` souboru definovat propojení pinů desky se signály v návrhu.

4.4.3 FITkit -> PC

Funkcí `FPGA_SPI_RW_A8_DN` přijme mikrokontroler data z SPI na zadané adrese a pošle je pomocí `term_send_hex` do sériového portu. Program v počítači vyčte data z otevřeného sériového portu.

5 Problémy a jejich řešení

Pro spuštění Playeru je někdy zapotřebí nastavit globální systémovou proměnnou PKG_CONFIG_PATH. Chybová hláška bude žádat instalované knihovny `.pc`, k nimž nemůže najít cestu.

```
$ export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/Apps/player/build/client_libs/libplayerc
```

```
error : Failed to load plugin stageplugin. error : libtool reports error: file not found
error : plugin search path: /home/pdr/Desktop/gazebo/stage/tutorial-src:.:/usr/local/lib/
error : failed to load plugin: stageplugin error : failed to parse config file xyz.cfg
driver blocks
$ export LD_LIBRARY_PATH=
```

```
$ /opt/Xilinx/11.1/common/lib/lin/libstdc++.so.6: version ‘GLIBCXX_3.4.11’ not found...
$ export LD_LIBRARY_PATH=
```

```
stage: error while loading shared libraries: libstage.so.3.2.2: cannot open shared object
file: No such file or directory
$ export LD_LIBRARY_PATH=/stage/Stage-3.2.2-Source/libstage/:$LD_LIBRARY_PATH
```

6 To do

Že aby se předešlo problémům - přepsat část, kde se otvírají porty FITkitu. Ty jsou teď zadané natvrdo.

Reference

[1] *FITkit* [online]. [cit. 2. září 2011]. Dostupné na: <http://merlin.fit.vutbr.cz/FITkit/>.

[2] *Player Project* [online]. 26 November [cit. 2. září 2011]. Dostupné na: <http://playerstage.sourceforge.net/>.

[3] *Player project package* [online]. [cit. 4. září 2011]. Dostupné na: <https://launchpad.net/ thjc/+archive/ppa>.

[4] *Rtv/Stage GitHub* [online]. [cit. 2. září 2011]. Dostupné na: <https://github.com/rtv/Stage>.

[5] *Stage INSTALL.txt* [online]. [cit. 4. září 2011]. Dostupné na: <https://github.com/rtv/Stage/blob/master/INSTALL.txt>.

[6] Owen, J. *Player/Stage Manual* [online]. [cit. 2. září 2011]. Dostupné na: <http://www-users.cs.york.ac.uk/ jowen/player/playerstage-manual.html>.