STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ

Vojtěch Boček

Brno 2011

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 18. Informatika

GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ

Autor: **Vojtěch Boček**

Škola: **SPŠ a VOŠ technická,**
Sokolská 1 602 00 Brno

Konzultant: **Jakub Streit**

Brno 2012

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně, použil jsem pouze podklady (literaturu, SW atd.) citované v práci a uvedené v přiloženém seznamu a postup při zpracování práce je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Brně dne: 6.3.2012 podpis:

Poděkování

Děkuji Jakubu Streitovi za rady, obětavou pomoc, velkou trpělivost a podnětné připomínky poskytované během práce na tomto projektu, a Martinu Vejnárovi za informace o jeho programátoru Shupito.

Dále děkuji organizaci DDM Junior, za poskytnutí podpory.

Také bych chtěl poděkovat panu profesorovi Mgr. Miroslavu Burdovi za všeobecnou pomoc s prací.

Tato práce byla vypracována za finanční podpory JMK.

Anotace

Cílem této práce bylo vytvořit uživatelské prostředí určené k parsování a zobrazování surových dat posílaných z mikrokontrolérů v robotech, digitálních sondách apod. Hlavní vlastností programu je modulárnost – rozdělení na podčásti určené ke specifickým úkonům (Terminál, grafický parser, vykreslování grafů).

Klíčová slova: parser, analýza dat, program

Annotation

Purpose of this labor is to create graphical user interface for parsing and displaying raw data sent from embedded devices, robots, digital probes and other devices which are using microcontrollers. Main feature of this application is modularity – it is divided to sub-sections designed for specific operations (Terminal, graphical parser, graph drawer).

Key words: parser, data analysis, program

Obsah

Úvod

Při stavbě robotů (například na soutěž Eurobot) jsem se setkal s problémem při zpracovávání dat z poměrně velkého množství senzorů, které robot obsahuje, a jejich přehledného zobrazování. Nenašel jsem žádný program, který by mi vyhovoval – k dispozici jsou pouze komerční aplikace, které stojí poměrně velké množství peněz, anebo aplikace které dokáží zobrazovat pouze v jednom formátu – typicky graf. Z tohoto důvodu jsem se rozhodl napsat vlastní program.

Popis rozhraní

Svůj program jsem pojmenoval "Lorris", je vytvořený v C++ a využívá Qt Framework(v4.7)[1], což je multiplatformní framework, který umožňuje spustit aplikaci na více systémech – testoval jsem na Debian Linux (Wheezy, 64bit) a Windows 7.
Program je navrhnutý jako modulární aplikace, aby mohl zastřešit několik samostatných částí, které však mají podobnou oblast použití.
Je licencovaný pod GNU GPLv3 a zdrojový kód je dostupný na Githubu[2].

Další použité knihovny

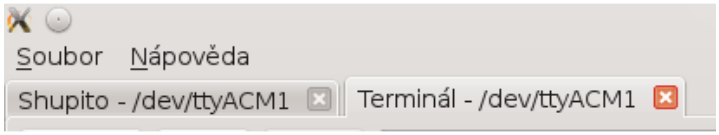
Qwt[3] je knihovna pro Qt Framework obsahující tzv. widgety pro aplikace technického charakteru – grafy, sloucové ukazatele, kompasy a podobně. Ve svojí práci zatím z této knihovny používám pouze graf (v modulu analyzéru), ale v budoucnu bych chtěl použít i některé další součásti.
QExtSerialPort[4] poskytuje připojení k sériovému portu a také dokáže vypsat seznam nalezených portů v počítačí.
QHexEdit2[5] je hex editor použitý v modulu programátoru Shupito na zobrazování obsahu paměti. V této knihovně jsem upravoval několik málo drobností, týkajících se především vzhledu.

Moduly

Každý modul se otevírá v samostatné záložce, přičemž je možné mít otevřeno více stejných modulů zároveň, a aby více modulů sdílelo jedno připojení (např. Teminál a Analyzér, oba pracující s jedním sériovým portem).

Současné moduly

- Analyzér – zobrazování dat v grafických widgetech, hlavní část práce.
- Proxy mezi Sériovým portem a TCP socketem – umožňuje vzdálený přístup k sériovému portu
- Shupito – rozhraní pro obsluhu programátoru "Shupito"[6]
- Terminál – terminál pro sériový port s podporou bootloaderu pro čipy ATmega[7].



Obrázek 1: Záložky modulů

Analyzér

Obrázek 2: Modul analyzér

Tento modul parsuje data přicházející sériovou linkou a zobrazuje je v grafických "widgetech". Zpracovaná data je možné uložit a později zase v programu otevřít. Předpokládá se, že data mají formát packetů.

Struktura dat se nastavuje v samostatném dialogu (viz obrázek 3), kde je možno nastavit délku packetu, jeho endianess[8], přitomnost hlavičky a její obsah – statická data ("start byte"), délka packetu(pokud je proměnná), příkaz a ID zařízení. Podle příkazu a ID zařízení je možno později data filtrovat.

Po nastavení struktury se přijatá data začnou po packetech zobrazovat v horní části okna, a v pravé části se zobrazí sloupeček s dostupnými zobrazovacími widgety. Widgety se dají pomocí drag&drop principu "vytahat" na plochu v prostřední části okna. Data se k widgetu přiřadí taktéž pomocí drag&drop, tentokrát přetažení prvního bytu dat na widget.

Poté widget zobrazuje data tohoto bytu, nebo tento byte bere jako první, pokud jsou data delší. Aby bylo možné zpětně poznat který byte je k widgetu přiřazen, je po najetí myši na widget červeně zvýrazněn.

Nastavení widgetu (jeho jméno, u čísla např. jeho datový typ apod.) jsou přistupná v kontextovém menu po pravém kliknutí myší na widget. Widgety je taktéž možné "uzamknout", aby nebylo možné je zavřít, měnit jejich pozici a velikost.

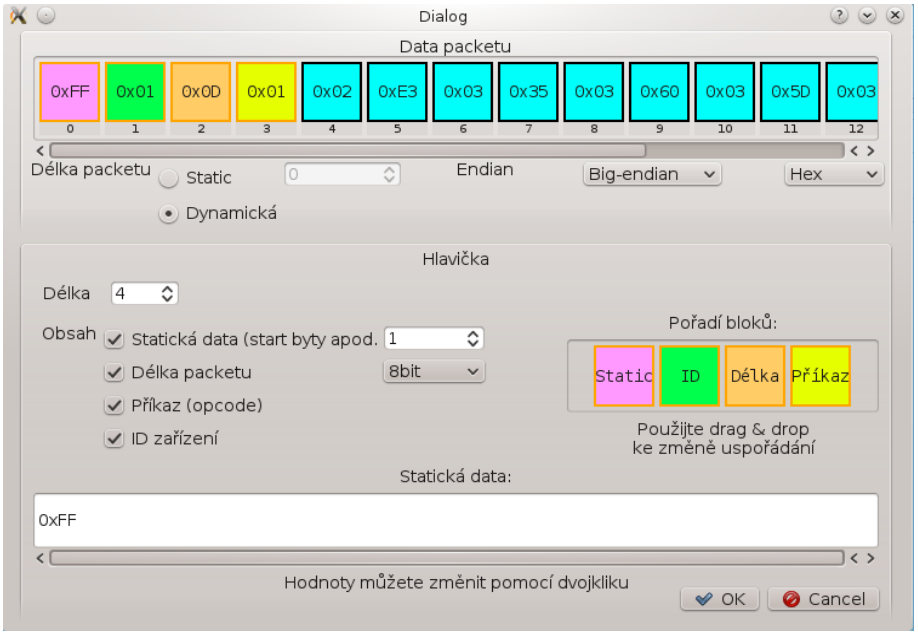
Aplikace si také přijatá data ukládá – navigace je umožněna posuvníkem a boxem v horní části okna.

Dostupné widgety

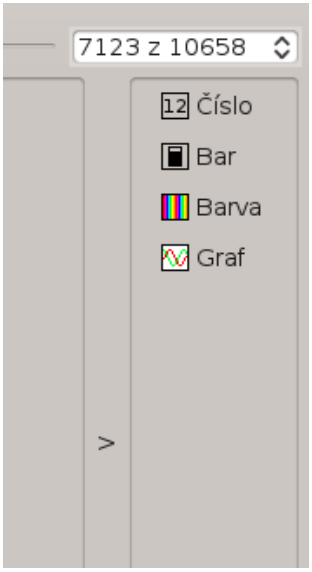
Číslo - Dokáže zobrazovat celá čísla (se znaménkem i bez, 8 až 64 bitů dlouhé) a desetiná čisla (single-precision[9], 32bit a 64bit – jako v jazyku C).
Widget dokáže zarovnat číslo na maximální délku jeho datového typu a formátovat ho těmito způsoby:

- Desítkový – číslo v desítkové soustavě
- Desítkový s exponentem – použije exponent pro zapsaní velkých čísel. Dostupné pouze pro desetiná čísla.
- Hexadecimální – výpis v šestnáctkové soustavě. Dostupné pouze pro typy bez znaménka.
- Binární – zobrazí číslo ve dvojkové soustavě

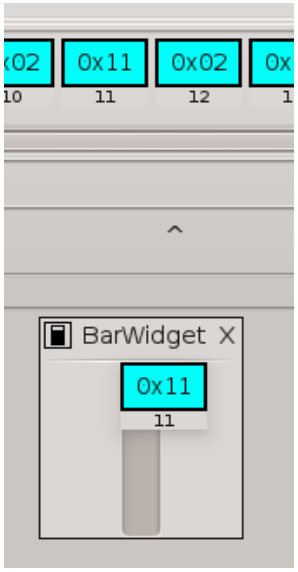
Bar - Sloupcový



Obrázek 3: Dialog nastavení struktury dat



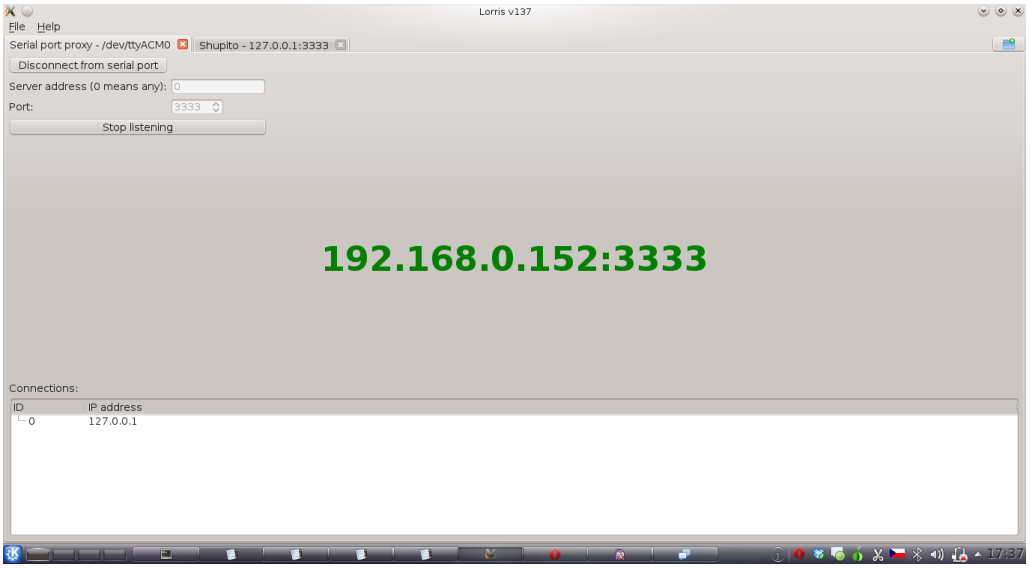
(a) Seznam widgetů



(b) Přiřazení dat pomocí drag&drop

Obrázek 4: Widgety

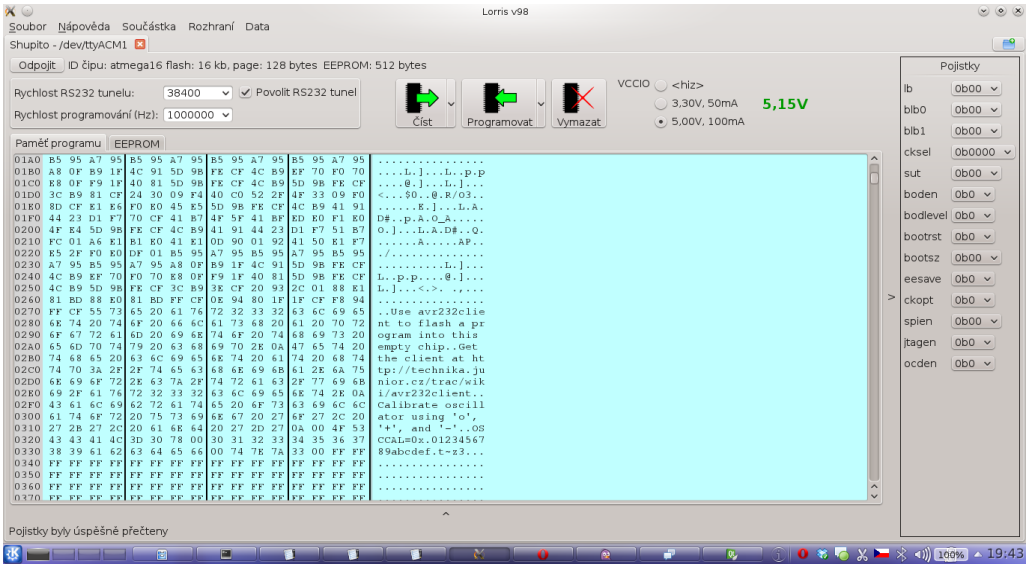
Proxy mezi sériovým portem a TCP socketem



Obrázek 5: Proxy mezi sériovým portem a TCP socketem

Jednoduchá proxy mezi sériovým portem a TCP socketem, která umožňuje vzdálené připojení na sériový port, buďto z Lorris nebo jiného programu.

Shupito

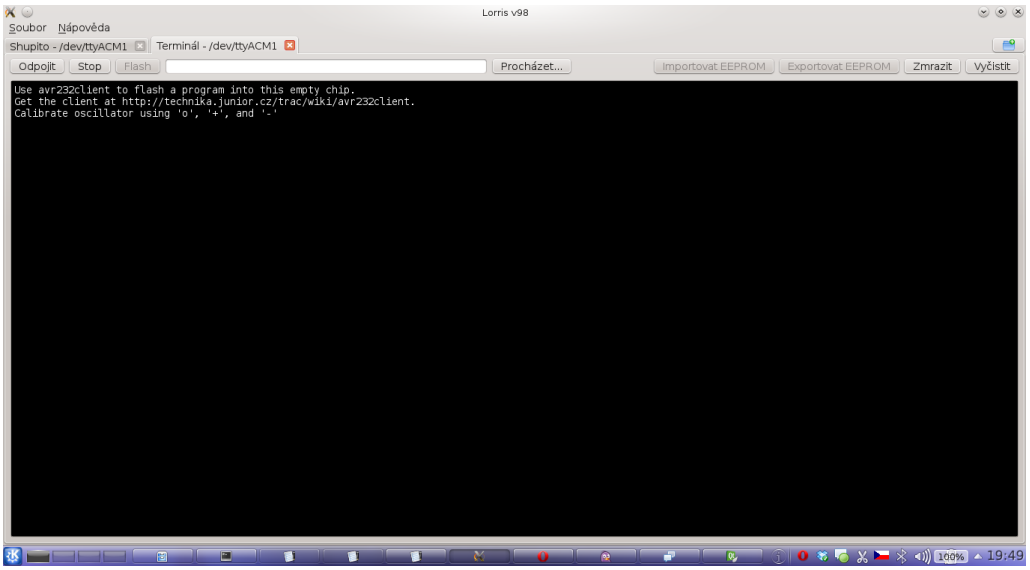


Obrázek 6: Modul Shupito

Shupito je programátor mikročipů vytvořený Martinem Vejnarem. Dokáže programovat mikrokontroléry pomocí ISP, PDI *DOP: zeptat se martina co všechno to zvládá*.

Modul v mojí práci dokáže programátor Shupito obsluhovat – nastavovat výstupní napětí, číst a programovat paměť čipů (flash i EEPROM) a číst a měnit pojistky. Jako výstupní i vstupní data používá soubory ve formátu Intel HEX32[10]. Shupito dokáže pracovat také jako RS232 tunel, i tuto funkci program podporuje – aktivní tunel se zobrazí jako další typ připojení a je možné ho využívat v ostatních modulech. Komunikace s programátorem je naportována z oficiálního ovládacího programu[7], který je však dostupný pouze pro MS Windows.

Terminál



Obrázek 7: Modul terminál

Klasický terminál – zobrazuje data přijatá přes sériový port a posílá stisky kláves, obohacený o podporu bootloaderu pro mikrokontroléry AVR ATmega(bootloader byl taktéž napsaný Martinem Vejnarem), který umožňuje jejich programování přes RS232 linku. Informace o protokolu bootloaderu jsem získal z oficiálního programu určeného k programování přes tento bootlodaer, avr232client[7].

Možnosti dalšího rozšiřování programu

Díky modulárnosti programu je možné poměrně jednoduše vytvořit další moduly, například modul pro ovládání robota přes bluetooth pomocí joysticku apod. Je také možné přidat další typy připojení – například socket, pokud jsou data nejdříve zpracována nějákým jiným programem.

Co se týče Analyzéru, je možné přidat další widgety – mám v plánu implementaci QtScriptu[11], který podstatně rozšíří možnosti parsování dat.

Reference

[1] *Qt Framework – Cross-platform application and UI framework*
http://qt.nokia.com/ (Stav ke dni 28.1.2012)

[2] *GIT repozitář Lorris*
https://github.com/Tasssadar/Lorris

[3] *Qt Widgets for Technical Applications*
http://qwt.sourceforge.net/ (Stav ke dni 22.2.2012)

[4] *Qt interface class for old fashioned serial ports*
http://code.google.com/p/qextserialport/
(Stav ke dni 22.2.2012)

[5] *Binary Editor for Qt*
http://code.google.com/p/qhexedit2/ (Stav ke dni 22.2.2012)

[6] *Shupito – Programátor*
http://shupito.net/ (Stav ke dni 28.1.2012)

[7] *avr232client*
http://technika.junior.cz/trac/wiki/avr232client
(Stav ke dni 28.1.2012)

[8] *Endianness*
http://en.wikipedia.org/wiki/Endianness (Stav ke dni 28.1.2012)

[9] *Single-precision floating-point format*
http://en.wikipedia.org/wiki/Single_precision (Stav ke dni 28.1.2012)

[10] *Intel HEX*
http://en.wikipedia.org/wiki/Intel_hex (Stav ke dni 28.1.2012)

[11] *QtScript*
http://developer.qt.nokia.com/doc/qt-4.7/qtscript.html
(Stav ke dni 28.1.2012)