

## Vizualizace hry v šachy v reálném čase pomocí desktop robota se čtyřmi stupni volnosti

Gavenda Tomáš · Elektrotechnika, Informačné technológie

18.08.2010



Účelem práce byla vizualizace hry v šachy pomocí polohování figur robotem JANOME JR 2200N. Robot je řízen osobním počítačem přes sériový port a zvládá po obdržení pozic figur samostatně figury přesouvat a vykonávat tak tahy hráče i protihráče. Program pro ovládání robota je vytvořen v jazyku C. Součástí práce je také vytvoření knihoven pro ovládání základních pohybů a funkcí robota. Řešení je obecné a je možné je využít i pro jiné účely než hru v šachy.

### 1. Úvod a zadání práce

Předem bych rád poděkoval panu doc. Ing. Mgr. Milanu Kvasnicovi Csc. za uvedení do problému, zadání úkolu a za podporu při řešení. Také děkuji panu Ing. et Ing. Eriku Královi za pomoc během práce a panu Ing. Petru Navrátilovi PhD. za cenné rady. Tuto práci jsem si vybral jako práci navíc za účelem zdokonalení se v oblasti robotiky a programování, byl jsem jenom rád, že mi bylo umožněno působit jako pomocná vědecká síla a doufám, že jsem splnil úkol lépe než dostatečně.

Zadáním bylo vytvořit funkční spojení mezi osobním počítačem a robotem za účelem vytvoření hry v šachy. Po zvážení možností byl vybrán desktop robot od firmy Janome, na kterém jsem jako pomocná vědecká síla tento rok pracoval a vytvořil program pro jeho obsluhu skrz sériový port a následně pro uskutečnění hry v šachy. Již původní návrh řešení počítal s použitím převzatého šachového enginu, resp. jeho zabudováním do programu, který jej umožní využít. Tak se i stalo a program využívá šachový engine Faile verze 1.4, který je volně dostupný, měnitelný a šiřitelný.

### 2. Výběr robota, programovacího jazyka

Již jsem předznamenal, že pro účely zadání byl vybrán robot od japonské společnosti Janome, konkrétně model JR2200N. Janome JR2200N je tzv. desktop (nebo také stolní) robot, pracující ve čtyřech osách, ale v poměrně úzce vymezeném prostoru (dáno konstrukcí robota). Tento typ byl zvolen převážně z důvodu jeho přenositelnosti a od této vlastnosti se odvíjející možnosti program předvádět mimo robotickou laboratoř. Janome je také mnohem bezpečnější ve srovnání s ostatními roboty, které škola vlastní (je velice obtížné dojít k úrazu při manipulaci s robotem).



Obr. 1: Robot Janome JR2200N.

Jako programovací jazyk byl vybrán jazyk C (C/C++) z důvodu variabilnosti a široké škály možností. Jazyk C byl zvolen také pro to, že jako student oboru BTSM mám znalosti právě tohoto jazyka na vyšší úrovni než znalosti programovacích jazyků jiných.

#### **Příslušenství potřebné pro vizualizaci hry**

Po výběru robota a programovacího jazyka bylo nutné sehnat či vyrobit odpovídající příslušenství pro vizualizaci hry. To znamená šachovnici, figury a uchopovací nástavce na čelist robota. Vše je řešeno pouze provizorně – tak, aby vše fungovalo, ale je poměrně vysoká pravděpodobnost, že bude docházet k chybám (ne ze strany robota). Nástavce na čelist jsou vyrobeny ze stavebnice Merkur, šachovnice a figury jsou ze dřeva. Šachovnice se sestává z černobílých polí a zásobníku figur, v nichž jsou otvory pro zasazování figur, což jsou dřevěné válečky jištěné a označené papírovými znaky.



Obr. 2: Šachovnice s figurami.

#### **4. Řešení zadání**

Nyní se dostáváme k samotnému řešení zadání práce. Postupně v podkapitolách rozvedu co se muselo řešit a jak. Jako první bylo nutné vyřešit komunikaci počítače a robota.

#### 4.1 Komunikace mezi počítačem a robotem - hardware

Veškerá komunikace se provádí po sériové lince. Robot má několik pracovních módů, a jelikož bylo nutné kompletně robota ovládat skrz počítač, byl zvolen mód „External run mode“. Bylo nutné také upravit několik vnitřních nastavení robota. Veškeré startovací impulsy byly nastaveny na COM port, a byly uvolněny všechny systémové vstupy a výstupy, tak, aby se dalo s robotem hýbat a neporušoval určitá vnitřní pravidla. Konkrétně se jedná o systémové vstupy, na které je připojena čelist robota, která by se, pokud by nebyly učiněny změny, při jakémkoliv pohybu sama zavírala.



Obr. 3: Spojené sériové porty počítače a robota.

#### 4.2 Komunikace mezi počítačem a robotem - software

Softwarové řešení komunikace je poměrně značná část celého programu, který byl pro hru a její vizualizaci vytvořen. Program pojmenovaný BetaOvladaniJ.exe, resp. jeho zdrojový soubor BetaOvladani.cpp, zahrnuje knihovnu Seriál.h, která je volně dostupná z internetu a pomocí standardní knihovny windows.h ovládá sériový port.

Data zasílaná z počítače do robota musí dodržovat určitý pevně daný a v manuálu popsany formát. Jedná se tzv. hexadecimální ASCII, neboli HEXASCII znaky, nikoliv binární znaky. Například 2 byty čísla „76“ se převedou na „004C“ v hexadecimálním tvaru. Veškerá data jsou pak posílána v pevně daném pořadí - nejprve počáteční znak přenosu - „\$“, dále pak kód příkazu (velké písmeno abecedy), kód podpříkazu, další údaje a na závěr kontrolní součet a prázdný znak klávesy Enter s ASCII hodnotou 13 pro ukončení zprávy. Znak 13 se velice špatně přidává na konec řetězce, vyřešil jsem to následující funkcí:

```

26 void posli(char* ddata)
27 {
28     int delka;
29     delka=strlen(ddata);
30     delka++;
31
32     sscanf(ddata,"%s",&data);
33     data[delka-1]=13;
34     data[delka]=0;
35
36     serial.Write(&data[0]);
37 }

```

Obr. 4: Funkce pro odeslání příkazu s přidáním znaku 13.

Tato funkce také rovnou odesílá příkaz robotovi (`serial.Write(&data[0]);`). To znamená, že je zajištěno, že každý odesílaný příkaz je ukončen automaticky a správně. V programu jsou vytvořeny funkce pro jednoduché příkazy, jako je otevření či zavření čelisti či inicializace robota. Ovšem pohyb robota je složitější. Přejdeme tedy k dalšímu bodu, což je výpočet souřadnic pohybu.

Tab. 1: Příklady příkazů.

Příkaz odesílaný robotovi	Význam příkazu
\$R082	Inicializace robota
\$K2000300000001C1	Otevření čelisti
\$K3000300000001C2	Uzavření čelisti

### 4.3 Výpočet souřadnic pohybu

Robot Janome se pohybuje v souřadnicovém systému, který je omezen velikostí pracovní plochy. Ta je velká 20 cm v ose x, 20 cm v ose y a 5 cm v ose z. Osu r (rotaci) neuvažují, protože nebylo zapotřebí ji využít. Ovšem robotovi nelze odeslat souřadnice jednoduše. Souřadnice se musí přepočítat při zachování pravidla, že se souřadnice násobí 2000, převede se na celé číslo, pak se zpět dělí 2000 a přepočítá na HEXASCII hodnotu. Takovýto údaj se pak připojí za příkaz pro pohyb robota z bodu do bodu (tzv. PTP) a nakonec se přidá kontrolní součet a příkaz je připraven k odeslání. Všechny tyto záležitosti řeší v programu funkce `prevod()`, `secti()` a `prevods()`.

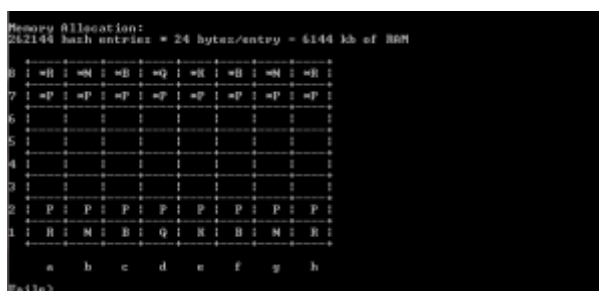
### 4.4 Funkce pro hru v šachy

Když jsou vyřešené funkce pro pohyb z bodu do bodu a otevření a zavření čelisti, je možné řešit další funkce pro samotnou hru v šachy. Tyto funkce jsou označeny jako `vezmi(z_kama_x, z_kama_y)`, `dej(kam_x, kam_y)`, `nachystej()`, `sachy()` a `sachyspoc()`. `Vezmi()` a `dej()` jsou pouhé sekvence předchozích funkcí - `vezmi()` otevře čelist, dorazí k figuře, resp. na její souřadnice, zavře čelist a zvedne figuru nad ostatní figury z šachovnice. `Dej()` funguje analogicky naopak. `Nachystej()` je podfunkcí funkcí `sachy()` a `sachyspoc()`. Má za úkol rozestavět figury ze zásobníku figur na pole šachovnice. Toto je umožněno vytvořením smyček `for()`, a volání funkce `vezmi()` a `dej()`.

`Sachy()` je již komplexní funkce, která má za úkol vizualizovat hru v šachy mezi dvěma lidskými soupeři v reálném čase. Předpokládá se znalost pravidel, a proto je řešena

pouze situace, kdy má být vyhozena figura, tzn. že robot nejprve přesune vyhazovanou figuru do zásobníku a poté teprve přesune vyhazující figuru na její místo. Ostatní hra je již jen na hráčích. Důležité bylo v této funkci vyřešit převod souřadnic zadávaných klasicky šachově („d2d4“) do souřadnic robota.

Sachypoc() pak spouští šachový engine Faile 1.4, který obsahuje veškerá pravidla a umělou inteligenci pro hru v šachy s počítačem v reálném čase. Ovládá se podobně jako sachy(), tj. pouze psaním souřadnic jako jsou např. „d2d4“. Navíc obsahuje možnost inicializovat robota zadáním slova „init“ (v případě, že by došlo k posunu motoru při nečekané chybě) a možnost ukončit hru příkazem „konec“. Do původního enginu bylo nutné doplnit veškeré funkce pro pohyb robota, pravidla pro vyhazování figur a v neposlední řadě pravidla pro rošádu.



Obr. 5: Šachový engine Faile 1.4.

#### 4.5 Hlavní programová smyčka

V hlavní programové smyčce se otevírá sériový port počítače, poté se volá funkce menu() a je na uživateli, co bude s robotem dělat. Nabízí se základní možnosti ovládání robota a spuštění šachových partií. Po ukončení programu se sériový port opět uzavírá.

```

38 void menu()
39 {
40     system("cls");
41     printf("Ovladani robota JANOME pres COM port\n");
42     printf("-----\n\n");
43     printf("Stisknete cislo cinnosti, kterou ma robot vykonat:\n");
44     printf("1 -> Inicializace (spusteni)\n");
45     printf("2 -> Start programu\n");
46     printf("3 -> Stop behem programu\n");
47     printf("4 -> Otvori celist\n");
48     printf("5 -> Zavri celist\n");
49     printf("6 -> Jdi na souradnice\n");
50     printf("7 -> Jdi na zacatek\n");
51     printf("8 -> Hra v sachy clovek vs. clovek\n");
52     printf("9 -> Hra v sachy clovek vs. pocitac\n");
53     printf("ENTER -> Konec programu\n"); // " -> \n"
54 }

```

Obr. 6: Funkce menu().

### 5. Závěr

Rád bych na závěr podotknul, že v této práci jsou obsaženy pouze stručné postupy práce a výsledky, protože byl stanoven limit stránek práce. Pro komplexní vysvětlení problematiky je nejlepší praktická ukázka práce s robotem a procházení zdrojových kódů programu.

Myslím si, že zadání jsem vyplnil v rámci možností úplně, existuje řada možností, jak řešení rozšířit o další funkce. Význam práce vidím v tom, že jsem dokázal ovládat robota Janome přes počítač po sériové lince bez nutnosti programu výrobce a dal jsem tak základ pro tvorbu programů, které robota využijí i pro jiné účely než vizualizaci

---

her.

## **Literatura**

1. JANOME Sewing Machine Co., Ltd.: Operation manual (COM Communication).  
JANOME Sewing Machine Co., Ltd., Japonsko, 2006.
2. REGIMBALD Adrien: Faile version 1.4. MIT, USA, 2000.

---

Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, Nad Stráněmi 4511, 760  
05 Zlín, Česká republika  
Studentská tvůrčí a odborná činnost STOČ 2010

---