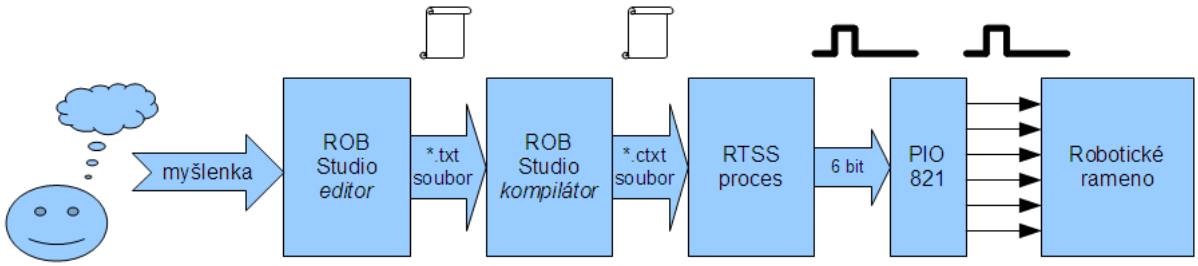
Ovládání robotického ramene

Základní koncepce

ROB 2-6 je robotické rameno s šesti stupni volnosti. Poloha kloubů je nastavována pomocí servomotorů, které jsou ovládány signálem, jež tvoří pulzy o frekvenci 50 Hz a o šířce 0,9 až 2,1 ms. Tyto pulzy jsou vysílány z data akviziční karty PIO-821 užitím digitálních výstupů, které jsou nastavovány a mazány v reálném čase RTSS procesem běžícím pod RTX. Vstupem do tohoto procesu je textový soubor s příponou ctxt, který obsahuje sekvenci příkazů, určující šířku pulzu do jednotlivých serv a dobu čekání mezi nimi.

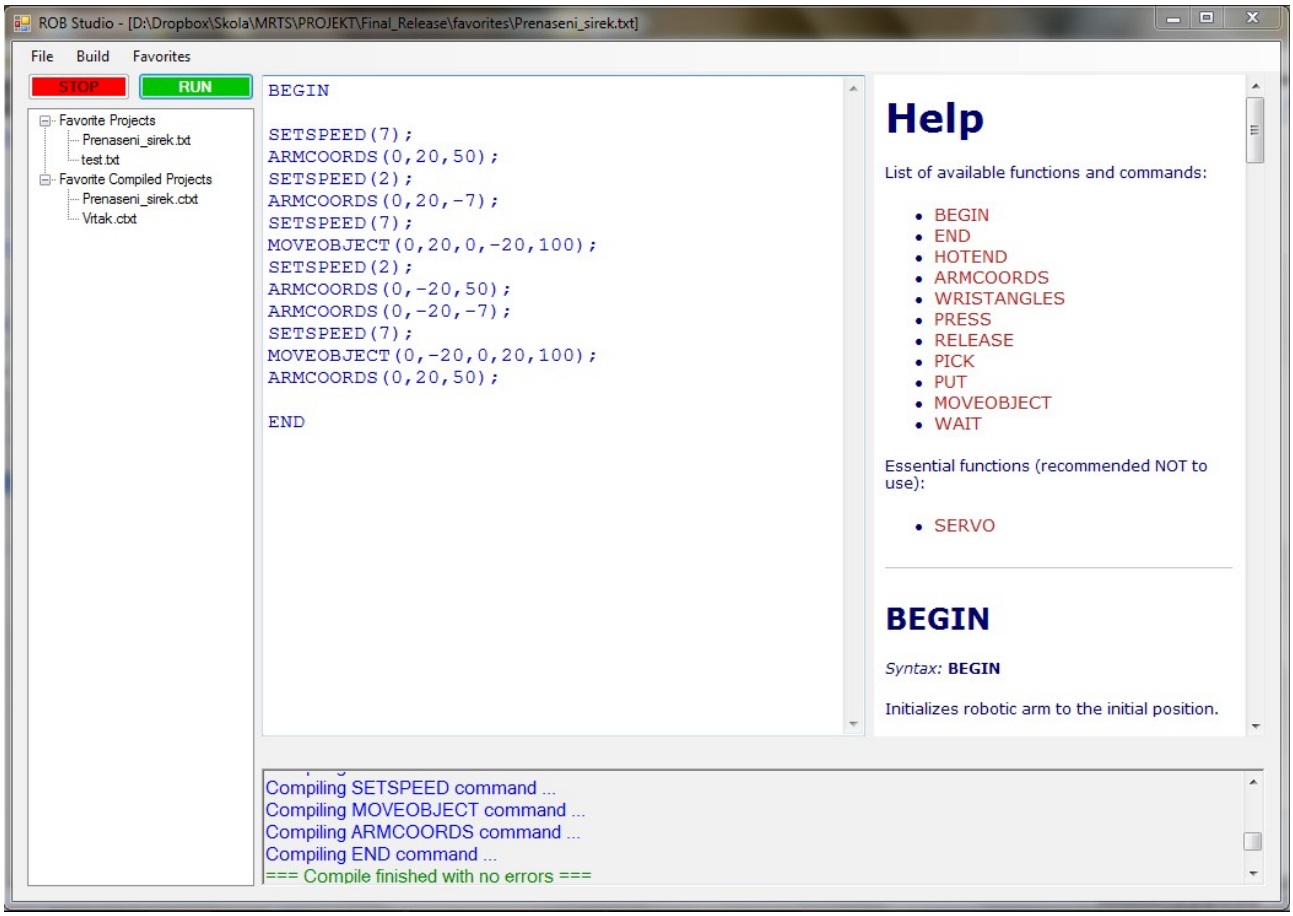
Protože užití těchto základních příkazů není příliš uživatelsky příjemné, bylo vytvořeno návrhové prostředí (ROB Studio), které poskytuje nástroj pro tvorbu zdrojových kódů ve specifickém skriptovacím jazyce, který již obsahuje uživatelsky příjemné funkce, pomocí kterých je možně jednoduše naprogramovat chování robota. Skript v tomto skriptovacím jazyce je kompilátorem ROB Studia přeložen do sekvence zákadních příkazů a uložen do souboru ctxt, který je pak zpracováván RTSS procesem.

Tato koncepce je zachycena na následujícím obrázku:



Zpět na obsah

Popis prostředí



V horní části programu je menu, které obsahuje tyto funkce:

- **File** - příkazy pro vytváření a otevírání textových souborů obsahující instrukce pro robotické rameno.
- **Build** - příkazy pro kontrolu syntaxe (Check Syntax), kompilaci do ctxt souboru pro RTSS proces (Compile) a spuštění zkompilovaného skriptu v RTSS procesu (Run)
- **Favorites** - příkazy pro přidávání nebo odebírání oblíbených souborů. Tyto soubory jsou uloženy ve složce favorites a jsou zobrazování ve stromové struktuře vlevo.

V levé části programu je stromová struktura oblíbených souborů.

V pravé části programu je nápověda obsahující popis všech příkazů, které skriptovací jazyk nabízí.

Ve spodní části programu je okno, které zobrazuje veškeré důležité hlášení a průběh jednotlivých operací.

Vlevo nahoře se nachází zelené tlačítko RUN, které plní stejnou funkci jako Build - Run (F5) a červené tlačítko STOP, které je aktivně pouze při běžícím procesu a umožňuje okamžité zastavení programu.

Zpět na obsah

Práce s prostředím

Postup práce s programem ROB Studio:

1. Vytvořte nový soubor (File - New Project ..., Ctrl+N)
2. Uložte soubor na vámi vybrané umístění (File - Save Project ..., Ctrl+S)
3. Napište do prostředního okna váš program a uložte jej (File - Save Project ..., Ctrl+S)

4. Spusťte program na robotickém rameni (Build - Run, F5 nebo zelené tlačítko RUN). Proběhnou všechny návrhové kroky: Check Syntax -> Compile -> Run. Pokud program obsahuje chyby, nebude zkompilován a spuštěn. Chcete-li provést pouze některé dílčí úkoly, použijte Check Syntax (F3) nebo Compile (F4)
5. Běžící program je signalizován ve stavovém okně. V případě nouze jej zastavte červeným tlačítkem STOP.

Zpět na obsah

Možnosti skriptovacího jazyka

Užitý skriptovací jazyk obsahuje dvě základní funkce, které umožňují nastavit přímo polohu všech šesti serv (SERVO a WAIT). Nad těmito funkcemi jsou pak další rozšiřující, které umožňují umístit koncový bod robotického ramena do libovolného bodu udaného v kartézských souřadnicích (ARMCOORDS) nebo natočit zápěstí do libovolné polohy a stisku (WRISTANGLES, PRESS, RELEASE). Kombinací výše uvedených funkcí pak vzniká poslední třída funkcí, která umožňuje přímo vyzvednout (PICK), umístit (PUT) nebo přesunout (MOVEOBJECT) objekt z daných souřadnic na jiné. V poslední řadě je možné také nastavit rychlost pohybu (SETSPEED).

Kompilátor řeší inverzní úlohu kinematiky pro výpočet polohy serv ze zadaných kartézských souřadnic a obsahuje také interpolační mechanismus, který zajišťuje, že se bude rameno mezi dvěma zadanými body pohybovat přímočaře.

Zpět na obsah

Jak to vše funguje

Uživatelský zdrojový kód je nejprve předán funkci CheckSyntax, která zeména za pomoci regulárních výrazů nejen odhalí syntaktické chyby, ale v mnoha případech také detekuje příčinu chyby. Déle kontroluje povolené rozsahy hodnot a syntetizovatelnost inverzní kinematické úlohy.

Pokud není odhalena žádná chyba, je text předán funkci Compile, která pro všechny zadané příkazy určí posloupnost bodů, do kterých je potřeba serva nastavit, včetně doby přechodu mezi nimi. Výsledek je předán interpolátoru, který vypočítá jakou rychostí se mají otáčet která serva, aby do stejného bodu dorazila současně.

Pokud systém úspěšně absolvuje kompilaci, dochází k běhu funkce Run, která, spustí RTSS proces, jehož parametr je jméno kompilovaného souboru.

RTSS proces je hlavní součástí aplikace. RTSS Proces nejprve nastaví prostředí a spustí časovač, který je spouštěn každých 20 ms, což odpovídá požadované periodě pulzů. Tento časovač nastaví příslušný pin do log. 1 a následně ve smyčce kontroluje, kolik od této události uplynulo času. Jakmile je dosaženo požadované šířky pulsu, je pin nastaven do log. 0. Druhé vlákno mezitím čte instrukce ze souboru a ukládá požadované hodnoty šířky pulsu do globální proměnné, odkud jsou čteny prvním vláknem.

Zpět na obsah

Výhody a nevýhody realizace

Program ROB Studio je napsán v jazyce C#, který společně s platformou .NET významně ulehčuje práci programátora. Díky tomu bylo možné v zadaném čase vytvořit aplikaci, která má všechny výše uvedené schopnosti. Nevýhodou však je, že pro toto prostředí není vytvořeno SDK, tudíž je možnost kontroly

aplikace omezená. V jazyce C# není možné používat RTAPI (alespoň ve verzi RTX 7.1) a sybsystém RTX má vlastní synchronizační objekty a události, které nejsou pomocí API funkcí Windows viditelné. Tento problém byl tedy vyřešen tak, že ROB Studio spustí RTSS proces pomocí služby RTSSrun a následně pomocí služby RTSSkill kontroluje a případně ukončuje běh tohoto procesu. Toto řešení není příliš elegantní, ale bohužel je v nejspíše za těchto okolností jediné možné. V případě užití v praxi by tak bylo nejspíše nutné přejít na vyšší verzi RTX nebo aplikaci přepsat do jazyka C++.

Zpět na obsah