

SIMULÁCIA SÚSTAV PRE S7PLCSIM V MATLABE

Igor Béla, Roman Král

Katedra automatizácie a regulácie, STU Bratislava, Fakulta elektrotechniky a informatiky

Abstrakt:

Príspevok obsahuje analýzu a opis prepojenia modelov simulovaných v reálnom čase v prostredí Matlab-Simulink so simulačným programom S7-PLCSIM, ktorý slúži na simuláciu PLC radu SIMATIC. Komunikácia medzi modelmi simulovanými v reálnom čase a programom S7-PLCSIM bude zabezpečená použitím ActiveX komponentu S7ProSim, ktorý bol vytvorený práve za týmto účelom. Uvedená aplikácia sa bude využívať v pedagogike na katedre automatizácie a regulácie, FEI STU Bratislava. V príspevku je uvedená stručná analýza problému, princíp komunikácie medzi modelom v Matlabe a riadením v S7-PLCSIM a opis realizácie prepojenia na strane Matlabu.

Kľúčové slová: Matlab, Real Time Toolbox, S-Funkcia, PLC, SIMATIC.

1 Úvod

Softvér S7-PLCSIM predstavuje jednoduché prostredie, v ktorom je možné rýchlo a jednoducho otestovať programy napísané pre PLC regulátory SIMATIC. Ide v podstate o simulačný program, ktorého úlohou je simulovať chovanie sa skutočného PLC s testovaným softvérom v reálnom čase. Avšak veľkou nevýhodou tohto systému je, že nedokáže priamo simulovať aj riadený systém. Tento problém možno riešiť rôznymi spôsobmi, z ktorých každý má svoje kladné aj záporné stránky.

Ako pomerne jednoduchý a spoľahlivý spôsob simulácie riadeného systému sa javí prepojenie simulácie procesu v reálnom čase v prostredí Matlab-Simulink s programom S7-PLCSIM pomocou ActiveX prvku S7ProSim. A práve týmto spôsobom sme sa zaoberali.

2 S7ProSim

S7ProSim je ActiveX komponent, ktorý vo vývojom prostredí poskytuje prístup k rozhraniu simulácie procesov S7-PLCSIM. Tento komponent môže byť použitý v každej aplikácii, ktorá umožňuje používanie ActiveX komponentov. Použitím tohto komponentu môžeme pripojiť simuláciu nejakého procesu k S7-PLCSIM-u. Tento ActiveX komponent poskytuje všetky metódy a udalosti, ktoré sú potrebné pre spoluprácu s S7PLCSIM-om.

Na Obr. 1 je znázornená postupnosť jednotlivých programových krokov využívajúcich S7ProSim tak, ako ich treba vykonávať pri komunikácii s S7-PLCSIM-om.

Jednotlivé programové kroky podrobnejšie:

Connect – pripojenie nášho programu k S7PLCSIM-u,

Enabled = True – týmto komponent zaregistrujeme v programe (sprístupnia sa tak niektoré udalosti),

ScanMode = SingleScan – odteraz sa bude snímanie hodnôt robiť jednotlivo na požiadanie a nie cyklicky,

WriteInputPoint() – zapíše konkrétnu hodnotu do obrazu výstupných premenných, ktoré budú neskôr vyslané do S7-PLCSIM-u,

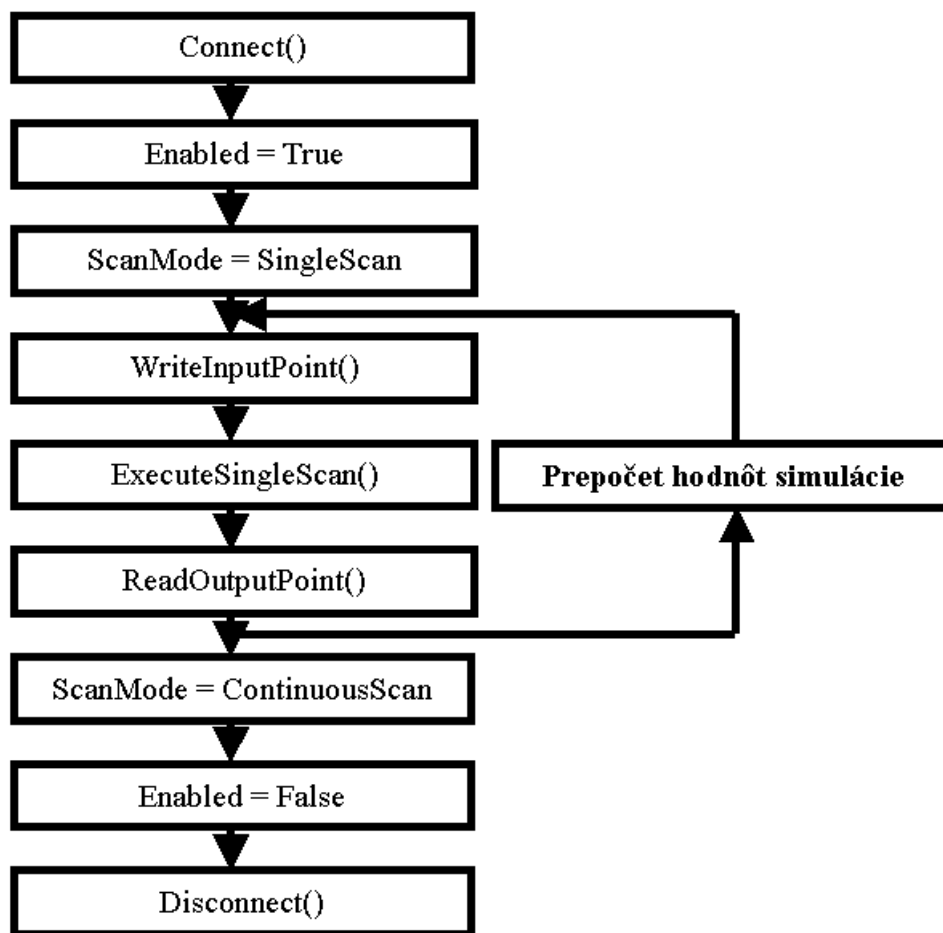
ExecuteSingleScan() – zapíše obraz výstupných premenných do S7-PLCSIM-u a zároveň načíta z S7-PLCSIM-u obraz vstupných premenných,

ReadOutputPoint() – načíta konkrétnu hodnotu z obrazu vstupných premenných,

ScanMode = ContinuousScan – snímanie hodnôt sa odteraz už nebude robiť na požiadanie ale cyklicky – riadenie prevezme S7-PLCSIM,

Enabled = False – komponent S7ProSim bude odregistrovaný z programu a niektoré udalosti už nebudú prístupné,

Disconnect() – odpojenie nášho programu od S7-PLCSIM-u.



Obr. 1: Postupnosť programových krokov pri programovaní s využitím S7ProSim-u

3 Možnosti komunikácie medzi prostredím Matlab-Simulink a externými aplikáciami pomocou ActiveX

Prostredie Matlab-Simulink má v sebe implementovanú podporu rozhrania ActiveX. To umožňuje v Matlab-Simulink-u používať rôzne interné a externé komponenty s podporou ActiveX a komunikovať pomocou ActiveX s rôznymi aplikáciami používajúcimi na komunikáciu štandard ActiveX.

Naším zámerom bolo vhodným spôsobom prepojiť prostredie Matlab-Simulink so simulačným programom S7PLCSIM pomocou ActiveX komponentu S7ProSim. Existuje niekoľko možných spôsobov ako to urobiť. My sme našli a vyskúšali nasledovné možnosti implementácie ActiveX komponentu S7ProSim:

- Implementácia priamo v simulačnej schéme vytvorenej v Simulinku.
- Implementácia v prostredí Matlab, avšak nie priamo v Simulinku s tým, že komunikácia medzi Matlab-om a Simulinkom bude zabezpečená iným spôsobom.
- Implementácia v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi Matlab-om a S7PLCSIMom s tým, že komunikácia medzi Matlab-om a Simulinkom bude zabezpečená iným spôsobom.
- Implementácia v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi Simulinkom a S7PLCSIMom.

V nasledovnom texte si jednotlivé možnosti rozoberieme podrobnejšie.

3.1 Implementácia ActiveX komponentu priamo v simulačnej schéme vytvorenej v Simulinku.

Implementácia ActiveX komponentu priamo v simulačnej schéme vytvorenej v Simulinku predstavuje pomerne elegantné riešenie. Je ho možné realizovať pomocou bloksetu Dials & Gauges, ktorý je voliteľnou súčasťou Matlab-u.

Prostredníctvom Dials & Gauges bloksetu máme možnosť ku štandardným ActiveX komponentom, dodaných spolu s bloksetom, pridať aj vlastný Active X komponent, ktorý nie je súčasťou dodávky. Takýmto spôsobom tam môžeme pridať aj S7ProSim.

Nevýhodou je, že tento blokset síce vie z voliteľných komponentov využívať ich ActiveX vlastnosti, ale nevie používať metódy, ktoré poskytujú. Keďže S7ProSim používa na komunikáciu takmer výhradne metódy, je ako voliteľný ActiveX komponent Dials & Gauges bloksetu v Simulinku prakticky nepoužiteľný.

3.2 Implementácia ActiveX komponentu v prostredí Matlab, no nie priamo v Simulinku

Ďalšou možnosťou je implementácia ActiveX komponentu v prostredí Matlab, avšak nie priamo v Simulinku s tým, že komunikácia medzi Matlab-om a Simulinkom bude zabezpečená iným spôsobom a to štandardnými komunikačnými prostriedkami využívanými na komunikáciu v Matlab-e, ako sú predávanie parametrov z Matlab-u do Simulinku pomocou funkcie **set_param** a výstupe simulovaných údajov zo Simulinku do Matlab-u.

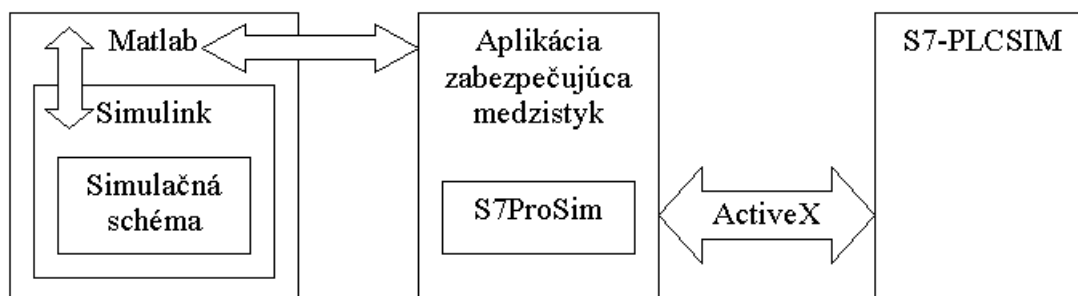
Komponent S7ProSim sa nám podarilo do Matlab-u importovať bez väčších problémov. V Matlab-e môžeme využívať aj metódy užívateľských ActiveX komponentov, čím vlastne vylúčime hlavný problém predchádzajúceho prístupu. Problém je ale v tom, že S7ProSim vo svojich metódach odovzdáva a prijíma parametre odkazom a nie hodnotou, čo v Matlab-e predstavuje dosť veľký problém vzhľadom k tomu, že Matlab niečo také ako odkaz alebo pointer nepozná. Tým pádom je aj tento prístup pre naše potreby nepoužiteľný.

3.3 Implementácia ActiveX komponentu v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi Matlab-om a S7PLCSIMom

V tomto prípade, tak ako aj v predchádzajúcom, musí byť štandardnými komunikačnými prostriedkami zabezpečená komunikácia medzi Matlab-om a Simulinkom. Na rozdiel od predchádzajúceho prípadu nebude ActiveX komponent S7ProSim importovaný v Matlab-e, ale v externej aplikácii. Tá bude s Matlab-om komunikovať buď cez ActiveX alebo cez nejaký iný komunikačný štandard podporovaný Matlab-om. Pomocou týchto štandardov možno z externých aplikácií jednoducho zadávať príkazy Matlab-u a preberať z Matlab-u výsledky týchto príkazov.

Tento spôsob komunikácie bol plne funkčný a odskúšaný aj s ActiveX komponentom S7ProSim. Je teda v našom projekte použiteľný.

Spôsob komunikácie je naznačený na obr.2.

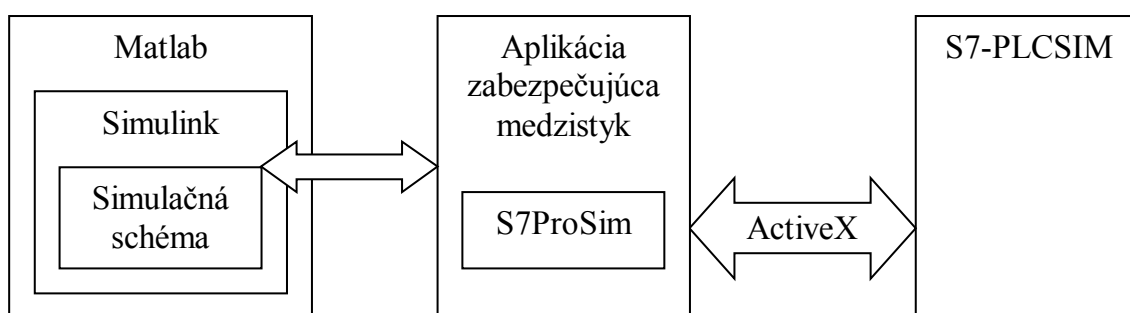


Obr.2: Spôsob komunikácie pri implementácii ActiveX komponentu v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi Matlab-om a S7PLCSIMom

3.4 Implementácia ActiveX komponentu v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi Simulinkom a S7PLCSIMom

Posledný prístup implementuje ActiveX komponent v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi S7PLCSIMom a priamo Simulinkom. Je zo všetkých štyroch prístupov najkomplikovanejší, pretože takýto spôsob komunikácie s externou aplikáciou nie je v Simulinku priamo podporovaný. Avšak to možno vyriešiť výberom vhodného spôsobu komunikácie medzi Simulinkom a externou aplikáciou. Najčastejšie sa na takúto komunikáciu využívajú simulinkové S-funkcie, pomocou ktorých možno implementovať kód využívajúci štandardné funkcie jazyka C a pri splnení niekoľkých podmienok aj Win32API funkcie. Takýmto spôsobom už možno zabezpečiť aj zložitejšiu komunikáciu – napr. TCP/IP.

Spôsob komunikácie je naznačený na obr.3.



Obr.3: Spôsob komunikácie pri implementácii ActiveX komponentu v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi Matlab-om a S7PLCSIMom

3.5 Zhrnutie

Vzhľadom k tomu, že prvé dva prístupy sú pre naše potreby nepoužiteľné a tretí je zbytočne komplikovaný a hlavne pomalý, rozhodli sme sa pre štvrtý prístup - implementáciu ActiveX komponentu v externej aplikácii, ktorá bude zabezpečovať medzistyk medzi Simulinkom a S7PLCSIMom. Ako komunikačné rozhranie medzi styčnou aplikáciou a Simulinkom sme sa rozhodli použiť rozhranie TCP/IP, ktoré má nasledovné vlastnosti:

- Obtiažnejšia implementácia vzhľadom k tomu, že Simulink rozhranie TCP/IP priamo nepodporuje.
- Vzhľadom na dnešné podmienky pomerne rýchla komunikácia.
- Možnosť rozdeliť zaťaženie počítača na viacero počítačov (Matlab ako server na jednom počítači, styčná aplikácia a S7-PLCSIM ako klient na druhom) – vyplýva z charakteru TCP/IP.

4 Možnosti zabezpečenia reálneho času a komunikácie s meracími kartami v simulačnom prostredí Matlab-Simulink

Aby simulácia v prostredí Matlab-Simulink prebiehala v reálnom čase, je nutné ju nejakým spôsobom zabezpečiť. Taktiež je nutné nejakým spôsobom spolupracovať s meracími kartami, ktoré by mali zabezpečovať styk s prípadnou reálnou sústavou napojenou na počítač, s ktorým pracujeme. Existuje mnoho prostriedkov ako to urobiť, tu sa však zameriame na Real Time Toolbox, ktorý dodáva česká firma HUMUSOFT.

Real Time Toolbox je nadstavba Simulinku ponúkajúca simulinkovú knižnicu blokov reálneho času, ktoré do Simulinku vnášajú simulačné prostredie reálneho času. Nadstavba obsahuje vstupy, výstupy a iné špeciálne bloky, ktoré umožňujú jednoduchý zber údajov, spracovanie signálov a riadenie aplikácií v reálnom čase bez potreby ďalšieho softvéru. Výkonné jadro zabezpečujúce v Real Time Toolbox-e reálny čas, podporuje vykonávanie

DSP operácií so vzorkovacou frekvenciou až 2 kHz. Real Time Toolbox obsahuje ovládače pre množstvo meracích kariet a teda nie je potrebné tieto ovládače programovať.

Výhody:

- Nie je nutné kompilovať danú simulačnú schému.
- Nie je nutné inštalovať C/C++ kompilátor.
- Win32API funkcie sú plne podporované.
- Práca s ním je rýchla a jednoduchá.

Nevýhody:

- Nie je veľmi rozšírený.
- Nemá takú širokú podporu meracích kariet ako napríklad Real-Time Windows Target.
- Pomalší ako RTWT – nevhodný na časovo kritické simulácie.

5 Rozhranie TCP/IP, jeho podpora a využitie v prostredí Matlab-Simulink v reálnom čase

Prostredie Matlab-Simulink síce poskytuje niekoľko priamych možností ako takúto komunikáciu zabezpečiť, avšak všetky majú jedno veľmi vážne obmedzenie - nepodporujú komunikáciu v reálnom čase. To je síce závažný problém, ale tento sa dá obísť nepriamymi metódami, ktorých je v prostredí Matlab-Simulink tiež niekoľko. V ďalšom texte sa budeme zaoberať práve týmito metódami.

Nepriame metódy komunikácie cez TCP/IP v prostredí Matlab-Simulink možno zadeliť do troch skupín:

- Metódy TCP/IP komunikácie, ktoré implementuje priamo Simulink + systém zabezpečujúci reálny čas (Real Time Windows Target, Real Time Toolbox, ...).
- Metódy TCP/IP komunikácie, ktoré implementuje prostredie Matlab-u a následne sa komunikácia medzi simuláciou v Simulinku a prostredím Matlab-u zabezpečuje internými komunikačnými prostriedkami Matlab-u.
- Metódy TCP/IP komunikácie, ktoré implementuje externá aplikácia mimo Matlab-u a následne sa komunikácia medzi Matlabom-Simulink-om zabezpečuje inými komunikačnými kanálmi (ActiveX, DDE, ...)

Výhody a nevýhody týchto metód sú opísané v [1]. Nás z týchto metód zaujala najmä prvá skupina, do ktorej patrí aj implementácia komunikácie cez TCP/IP pomocou simulinkovej S-funkcie. Práve pre tento spôsob komunikácie sme sa nakoniec aj rozhodli.

Implementácia simulinkovej S-funkcie, ktorá dokáže zabezpečiť komunikáciu cez TCP/IP bola podrobne popísaná v [1]. Išlo tu o jednoduchú S-funkciu implementujúcu TCP/IP server (prípadne klient), ktorá dokázala jednosmerne z Matlab-u v reálnom čase posielať namerané dáta vonkajšiemu klientovi. Túto funkciu sme prispôbili nasledovným požiadavkám:

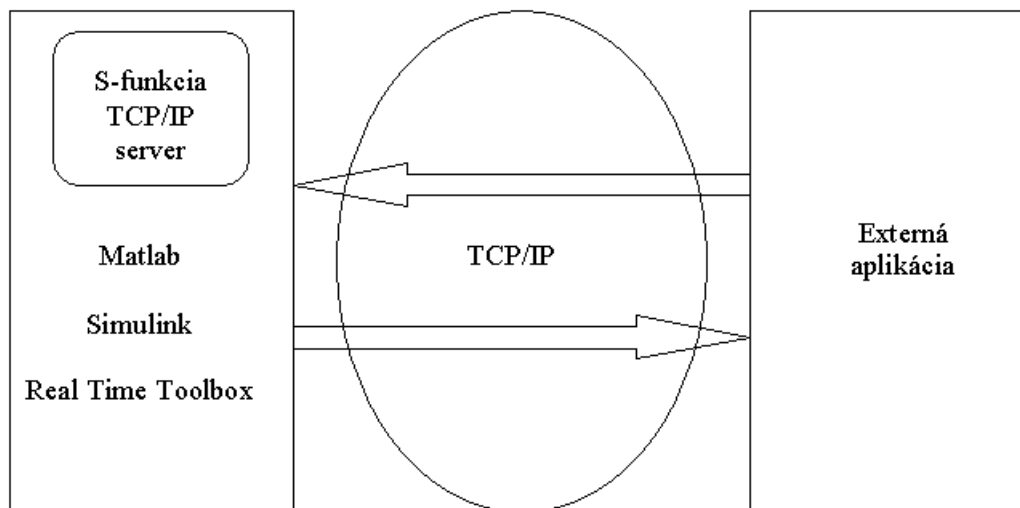
- Možnosť nadviazať komunikáciu cez TCP/IP a teda zabezpečiť funkciu servera.
- Možnosť posielať dáta zo servera (S-funkcia v Matlab-e) klientovi (externá aplikácia) cez TCP/IP.
- Možnosť prijímať dáta od klienta.
- Zabezpečiť komunikáciu v reálnom čase.
- Schopnosť korektne ukončiť svoju činnosť a oznámiť klientovi ukončenie činnosti.

Z týchto požiadaviek bolo jasné, že od servera v Matlab-e (S-funkcia) požadujeme obojsmernú komunikáciu. Pôvodný zámer bol upraviť S-funkciu z [1] pre Real-Time Windows Target tak, aby bola schopná nielen odosielať dáta klientovi, ale ich od klienta aj prijímať. To sa však ukázalo ako nemožné, pretože sme na TCP/IP komunikáciu nemohli použiť nevyhnutné Win32API funkcie. Tie mali zabezpečovať práve komunikáciu cez TCP/IP.

Tento problém sme však úspešne vyriešili použitím Real Time Toolbox-u, ktorý nemá s Win32API funkciami žiadny problém. Navyše, za predpokladu časovo menej náročnej simulácie, je použitie Real Time Toolbox-u v našom projekte takmer ideálne, pretože sme tým vylúčili najpodstatnejšiu nevýhodu Real Time Toolbox-u – nevhodnosť na rýchle simulácie.

Zámenou Real Time Toolbox-u za Real-Time Windows Target vo formulácii problému boli odstránené všetky prekážky, ktoré stáli v ceste vytvoreniu konkrétnej S-funkcie. Mohli sme teda pristúpiť k implementácii samotnej S-funkcie, plniacej úlohu TCP/IP servera.

Princíp komunikácie medzi S-fukciou a externou aplikáciou je naznačený na obr.4.

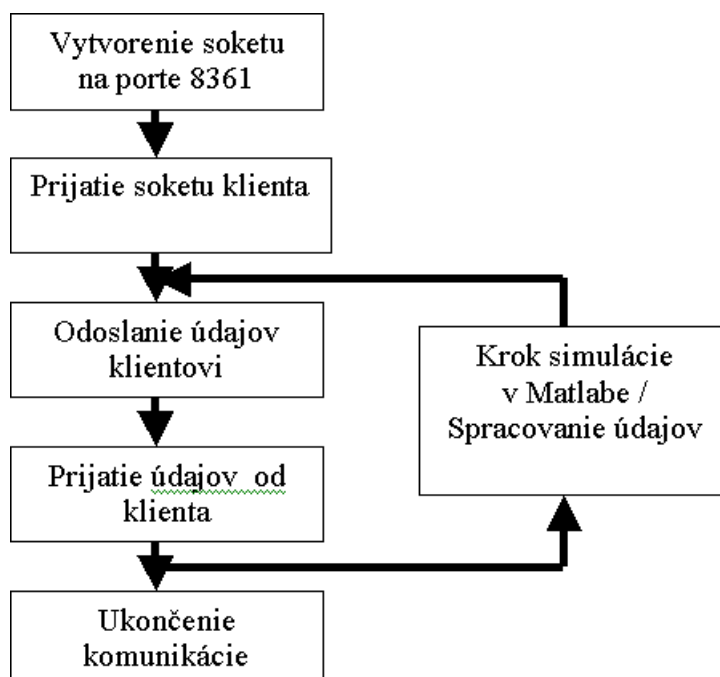


Obr. 4: Komunikácia medzi S-fukciou a externou aplikáciou cez TCP/IP

6 Realizácia prepojenia prostredia Matlab-Simulink s programom S7-PLCSIM

Táto časť opisuje S-funkciu zabezpečujúcu komunikáciu cez TCP/IP na strane Matlab-u. Pretože na strane Matlab-u plní funkciu servera, nazvali sme ju MTPserver.

Samotná implementácia S-funkcie MTPserver v prostredí Matlab + Simulink + Real Time Toolbox je principiálne jednoduchá. Princíp činnosti je naznačený na obr.5.

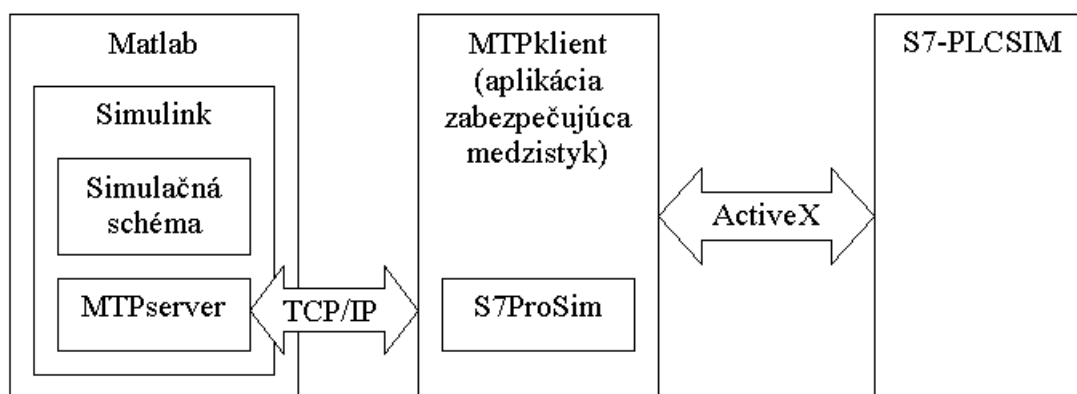


Obr. 5: Princíp činnosti S-funkcie MTPserver

Funkciu MTPservera možno zhrnúť v nasledovných bodoch:

- Všetky operácie so soketmi sú blokujúce.
- Po spustení servera sa vytvorí soket na porte 8361, ktorý čaká na pripojenie klienta.
- Ihneď po pripojení klienta začne komunikácia s klientom.
- TCP/IP server posiela dáta klientovi v jednotlivých krokoch simulácie v čase, ktorý mu na to prideli Matlab-Simulink.
- V tom istom časovom úseku, hneď za vyslaním dát klientovi, prijme server dáta od klienta.
- Keďže operácia čítania z klienta je blokujúca, klient musí obratom odoslať dáta serveru, aby v simulácii nebol narušený charakter reálneho času.
- Prebehne spracovanie údajov v Matlabe – Simulinku – RTT – zápis/čítanie do/z meracích kariet, úprava parametrov v Matlabe/Simulinku.
- Ak bol klientom vyslaný signál na ukončenie simulácie, simulácia skončí a klientovi sa odošle potvrdenie ukončenia.

Spôsob komunikácie medzi S-funkciou MTPserver a klientskou aplikáciou MTPklient je naznačený na obr.6.



Obr. 6: Princíp komunikácie medzi S-funkciou MTPserver a klientskou aplikáciou MTPklient

Keďže tento program plní pri TCP/IP komunikácii úlohu klienta, nazvali sme ho MTPklient. Funkciou tohto programu je zabezpečiť rozhranie medzi Matlab-om a S7-PLCSIM-om.

Aplikáciu MTPklient možno z funkčného hľadiska rozdeliť na dve časti:

- Hlavná časť aplikácie, ktorá má za úlohu obsluhovať udalosti prichádzajúce z užívateľského rozhrania a od systémových časovačov. Tiež zabezpečuje vlastnú komunikáciu a synchronizáciu s programom S7-PLCSIM a to pomocou komponentu S7ProSim.
- Vedľajšia časť aplikácie tvorená systémovým vláknom. Toto vlákno je použité iba na TCP/IP komunikáciu s S-funkciou v Matlab-e a nič iné.

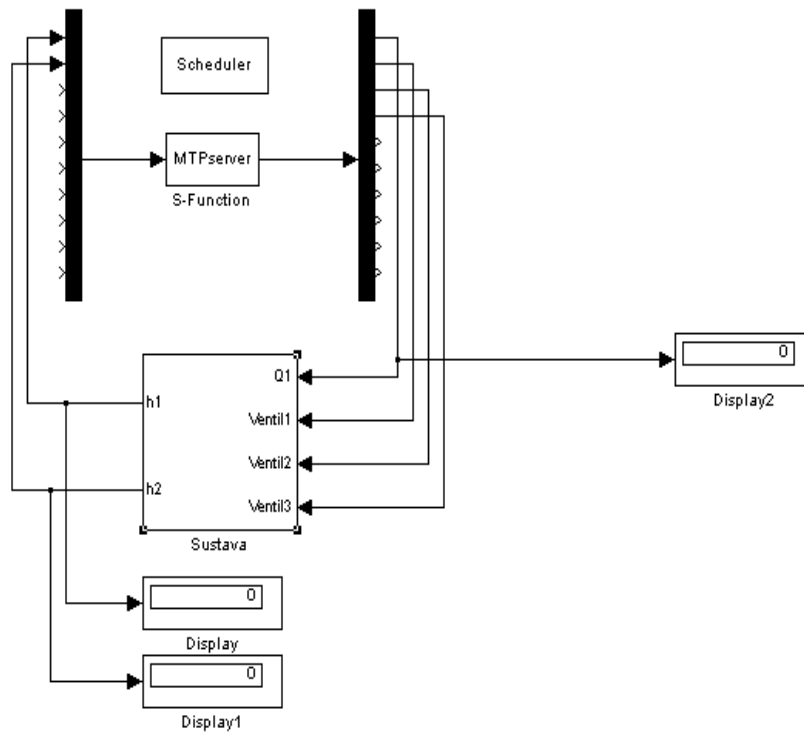
Obe tieto časti sú teda navzájom funkčne a časovo nezávislé, čo je dôležité vzhľadom na zabezpečenie reálneho času tak v Matlab-e ako aj v S7-PLCSIM-e. Synchronizácia a komunikácia medzi týmito časťami aplikácie MTPklient je realizovaná pomocou zdieľaných premenných, ku ktorým môžu rovnako pristupovať obe časti aplikácie.

7 Príklad aplikácie

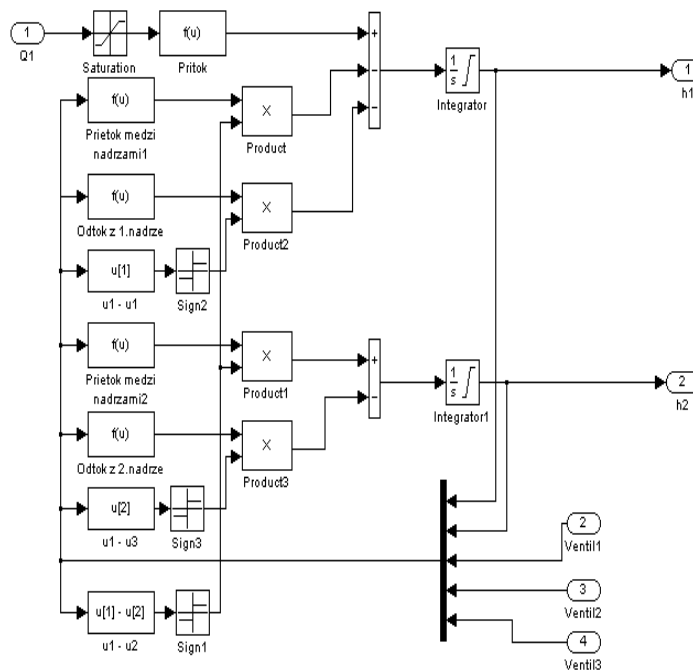
Opisované riešenie sme prakticky overili na príklade simulácie spojených nádob naplnených kvapalinou, pričom na vytvorenie serverovej a klientskej strany sme použili osobitne dva počítače v lokálnej sieti.

Na jednom bola v Matlabe-Simulinku ako súčasť simulovanej schémy implementovaná S-funkcia MTPserver a na druhom počítači bol program S7PLCSIM spolu so samostatnou aplikáciou MTPklient realizujúcou medzistyk.

Simulačná schéma realizovaná v Matlabe-Simulinku je znázornená na obrázkoch 7 a 8.



Obr. 7: Simulačná schéma s blokom S-funkcie MTPserver



Obr. 8: Simulačná schéma riadenej sústavy

Zhodnotenie a záver

Výsledkom práce je S-funkcia MTPserver a program MTPklient, ktoré zabezpečujú spojenie a komunikáciu medzi prostredím Matlab-Simulink a programom S7-PLCSIM. Všetky tieto programy využívajú na komunikáciu s programom S7-PLCSIM ActiveX prvok S7ProSim.

I keď nebolo pôvodne zamýšľané riešiť komunikáciu cez TCP/IP rozhranie, toto sa ukázalo ako dobrá voľba, ktorá priniesla do riešenia viaceré výhody.

I keď všetky spomínané programy sú dokončené, nie je vylúčený ich ďalší vývoj. Ten by mohol riešiť komunikáciu aj v binárnych hodnotách (nie iba hodnôt typu WORD), zväčšenie počtu premenných prenášaných medzi Simulink-om a PLCSIM-om (tu iba 10+10). Ďalej by bolo možné tu opísaným spôsobom realizovať riadenie reálneho procesu pomocou simulovaného PLC.

Pri vývoji bol používaný Matlab 6.1 (R12), RealTime Toolbox 3.1, Microsoft Visual Studio 6.0 + SP6, Borland C++ Builder 4 Standard, SIEMENS Simatic Step7 v5.1 + SP6 a S7-PLCSIM v5.0.

Použitá literatúra

- [1] Antal, D.: Riadenie reálneho modelu žeriavu v prostredí Internetu, Diplomová práca, KAR FEI STU Bratislava, 2002
- [2] Masár, I., Ivanov, I.: Aplikácie reálneho času v programovom prostredí MATLAB/SIMULINK. Slovenská technická univerzita, Bratislava, 2001.
- [3] Král, R.: Simulácia sústav pre S7PLCSIM. Záverečný projekt, KAR FEI STU Bratislava, 2003
- [4] Siemens Simatic S7ProSim Manual. Siemens 2001.
- [5] <http://www.mathworks.com> - oficiálna stránka firmy MathWorks, ktorá vyvíja prostredie Matlab-Simulink
- [6] <http://www.humusoft.cz> - oficiálna stránka firmy HUMUSOFT, ktorá vyvíja Real Time Toolbox
- [7] <http://www.mathtools.net> - portál zameraný na technical computing

Kontaktná adresa

Igor Béla, Katedra automatizácie a regulácie, Fakulta elektrotechniky a informatiky, STU, Ilkovičova 3, 812 19 Bratislava, Slovensko. E-mail: ibelai@kar.elf.stuba.sk.