

S-FUNKCE VYUŽÍVAJÍCÍ NÍZKOÚROVŇOVÝ PŘÍSTUP K FUNKCÍM MĚŘICÍ KARTY MF624

Š. Ožana

Vysoká škola báňská-Technická univerzita Ostrava

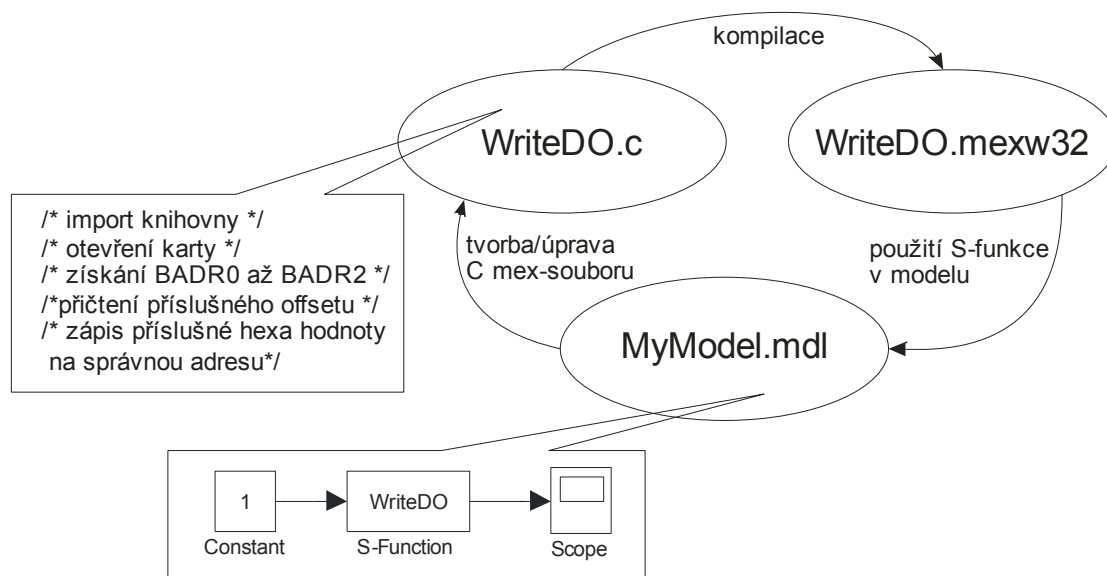
Abstrakt

Tento příspěvek se zabývá možností nízkourovňového programování měřicí karty MF624. Využívá koncepce S-funkcí v Simulinku, konkrétně C mex-souborů. Hlavní myšlenkou a přínosem je definovat S-funkce, kterých by bylo následně možno využít do Simulinkových modelů. Tímto způsobem by byly zdefinovány bloky funkcí, které nejsou obsaženy ve standardně dodávaném softwaru (Real Time Toolbox). Zejména se jedná o definování C mex S-funkcí funkcí, které by umožňovaly pracovat s čítači/časovači a plně by tak využily všech jejich možností.

1 Úvod

Čítače/časovače mohou pracovat v různých módech popsaných v kapitole 3-Programming Guide v manuálu k měřicí kartě MF624. Za všechny možnosti bych zmínil možnost měření času a frekvence s využitím programovatelných vnitřních hodin až do frekvence 50 MHz nebo možnost externího trigrování od náběžných či sestupných hran signálu. Typickou úlohou, ve které je takový přístup vhodný či nutný, je měření střidy periodických obdélníkových pulsů, nebo měření doby návratu vyslaného pulsu (signálu “burst”) z vysílače ultrazvukového snímače k přijímači. Tento přístup je obecně vhodný pro úlohy, ve kterých je zapotřebí přesného měření času (frekvence), například pro regulaci soustav s malými časovými konstantami (rychlý přechodový děj vyžadující rychlý akční zásah), například vzduchová levitace nebo inverzní kyvadlo.

Výchozím bodem je použití knihovny hudaqlib.dll dodávané s měřicí kartou. V této knihovně se nachází většina funkcí pro práci s I/O a čítači. Některé funkce/módy, zejména pro časovače, však musejí být naprogramovány za použití nízkourovňového přístupu, tj. zapsáním příslušné hodnoty na danou adresu. Zjednodušené schéma vývoje a použití těchto S-funkcí je znázorněno na obr. č. 1. Vytvořený a upravený zdrojový C mex-soubor (dodržující standardní strukturu mex-souboru), obsahující v těle programu funkce přímého zápisu, je zkompilován v příkazovém řádku Matlabu (nutno použít Visual C++ překladač). Jméno výsledného souboru s příponou *.mexw32 je poté použito coby název pro blok “S-Function” z knihovny User-Defined Functions. Výhodou uvedeného přístupu je to, že jej lze rovněž použít v prostředích C/C++/C#.



Obr. 1: Vývoj a použití S-funkce využívající nízkourovňový přístup k funkcím měřicí karty MF624

2 Demonstrační úloha č.1 – zápis na digitální výstupy

Uvedený kód zapíše sekvenci logických jedniček a nul na digitální výstupy (10101010 - tj. AA v hexa soustavě). Na začátku kódu se nejdříve nadefinují potřebné proměnné:

```
int h;
const HudaqResourceInfo* res;
int i;
bool a;
unsigned char *prom;
h = HudaqOpenDevice("MF624", 1, 0);
res = HudaqGetDeviceResources(h);
```

V části kódu uvedené “*static void mdlInitializeSizes(SimStruct *S)*” této C mex funkce poté následuje tento kód, který zapíše na BADR1+0x10 hodnotu AA, přesně dle manuálu ke kartě MF624:

```
static void mdlInitializeSizes(SimStruct *S) {
prom=res->MemResources[1].Base;
prom=prom+0x10;
*prom=0xAA;
.....
}
```

3 Demonstrační úloha č.2 – měření střídy

Uvedené části kódu ukazují, jak pracovat s časovači. Úkolem je stanovit střídu externě přiváděného signálu na svorkovnici X2. Řešením je nastavení čítače do módu gate. Čítač pak načítá pulsy od vnitřních hodin po dobu, kdy je přivedený signál v úrovni log1.

```
int h;
const HudaqResourceInfo* res;
int i;
bool a;
unsigned char *BADR2;
unsigned char *CTR1MODE;
unsigned char *CTR1CTRL;
unsigned int p1=0;

% mdlInitializeSizes
static void mdlInitializeSizes(SimStruct *S){
h = HudaqOpenDevice("MF624", 1, 0);
res = HudaqGetDeviceResources(h);
for (i=0; i<res->NumMemResources; ++i)
BADR2=res->MemResources[2].Base;
.....
}
```

% mdlOutputs

```
static void mdlOutputs(SimStruct *S, int_T tid) {
unsigned int a1,b1,c1,d1;
unsigned int pom1;
```

```

CTR1MODE=BADR2+0x10; //nastaveni prvnioho byte ridicoho registru
*CTR1MODE=0x01;
CTR1MODE=BADR2+0x11; //nastaveni druheho byte ridicoho registru
*CTR1MODE=0x28;
CTR1MODE=BADR2+0x12; //nastaveni tretioho byte ridicoho registru
*CTR1MODE=0x00;
CTR1MODE=BADR2+0x13; //nastaveni ctvrteho byte ridicoho registru
*CTR1MODE=0x00;

```

```

p1 = p1+1;
if(p1 >= 1000)
{
    p1 = 0;
    a1 = BADR2[0x14];
    b1 = BADR2[0x15];
    c1 = BADR2[0x16];
    d1 = BADR2[0x17];
    b1 *=256;
    c1 *= (65536);
    d1 *= (16777216);
    pom1=(a1+b1+c1+d1);
    *vystup1=pom1;//vystup casovac 1
    CTR1CTRL=BADR2+0x61;
    *CTR1CTRL = 0x02; // reset counter1.... nastaví 9. bit
}
}
}
% mdlTerminate

```

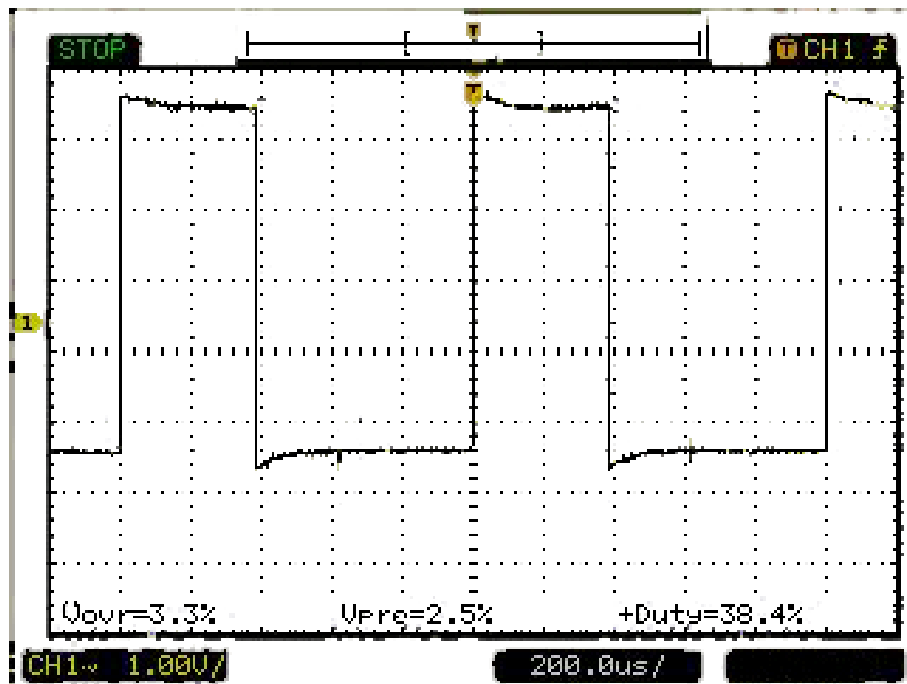
```

static void mdlTerminate(SimStruct *S)
{ HudaqCloseDevice(h); }

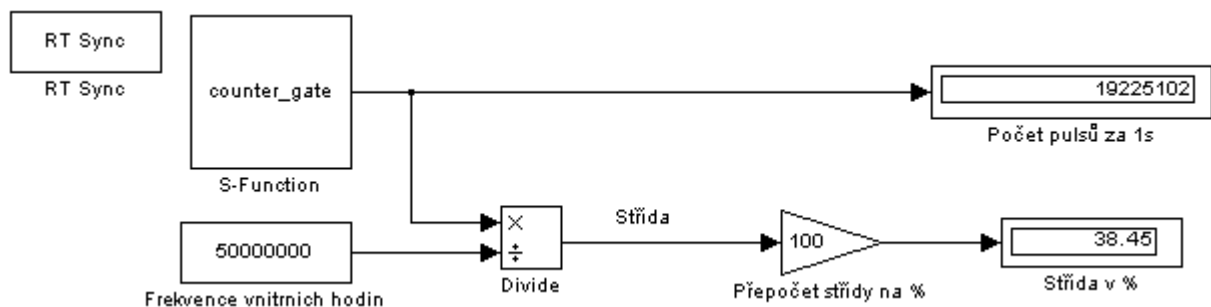
```



Obr. 2: Testovací signál přiváděný na časovač svorkovnice X2



Obr. 3: Ověření testovacího signálu na osciloskopu



Obr. 4: Simulinkové schéma pro měření střídy

4 Závěr

Tento příspěvek popisuje rozpracovaný stav studentského projektu nazvaného “Výzkum v oblasti řízení real-time aplikací” na katedře Měřicí a řídicí techniky při VŠB-TUO. Jedním z prvotních podnětů pro low-level programování karty MF624 byla úloha měření teploty pomocí polovodičového převodníku teplota/střída SMT160, která je již plně funkční. Aktuálně je dále zpracovávána úloha měření vzdálenosti pomocí ultrazvukového vysílače a přijímače UST40R a UST40T, při které bude vyhodnocována doba mezi odesláním a přijetím odraženého signálu. Stejný přístup bude použit pro ovládání a řízení dalších fyzikálních modelů v laboratoři, a to v případech, ve kterých je zapotřebí časovačů/čítačů pro zpracování TTL signálů, jako je například trigování nebo jakékoliv reakce na náběžné či sestupné hrany signálů za využití všech dalších možností, které časovače na kartě poskytují. Tento příspěvek a výše uvedený studentský projekt jsou sponzorovány společností ČEZ.

Reference

[1] <http://www2.humusoft.cz/www/datacq/manuals/mf624um.pdf>