VYSOKÉ UČENÍ TECHNICKÉ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dokumentace k projektu do předmětu IMP

MSP430 Výpočet odmocniny

2015/2016

Roman Halík (xhalik01)

1 Úvod

Tato dokumentace popisuje jednoduchý program pro FitKit, jehož funkcionalitou je výpočet odmocniny z celého čísla. V programu jsou pro výpočet implementovány dva algoritmy, konkrétně Newtonova (Babylonská) metoda, a Metoda půlení intervalů. Budou také komentovány charakteristiky obou algoritmů, ovládání aplikace a případné rozdíly v časové náročnosti a přesnosti těchto implementovaných algoritmů. Aplikace podporuje výpočet odmocniny z čísla v rozsahu datového typu int (16 bitů).

2 Ovládání

Stiskem jakékoli číselné klávesy je možné postupně napsat číslo, ze kterého má být vypočtena odmocnina, přičemž čísla se průběžně zobrazují na LCD. Výpočet se potvrdí stiskem tlačítka **A** nebo **B**, na která jsou namapovány algoritmy pro výpočet. Pro klávesu **A** je to Newtonova metoda, pro klávesu **B** metoda půlení intervalů. Stisk jakékoli jiné klávesy vede na neplatný vstup. Na tuto skutečnost je uživatel upozorněn odpovídající chybovou hláškou, která je zobrazena pouze na dobu nezbytnou pro její přečtení. Poté program přejde do výchozího stavu – lze tedy zadat nové číslo. Stejně tak není povoleno vkládat číslice větší, než je rozsah datového typu int (16 bitů). Takový vstup je anulován, a uživatel je na překročení rozsahu upozorněn.

3 Schéma zapojení, nastavení

Pro spuštění projektu na platformě FitKit nejsou třeba žádná zvláštní nastavení, ani zapojení externích zařízení.

4 Řešení

Program vychází z demo programu „Klávesnice a LCD“, jehož autorem je Ing. Zdeněk Vašíček, Ph.D. Zdrojové kódy byly upraveny pro potřeby projektu. Konkrétně se jedná o úpravy v souboru main.c, jehož obsah byl změněn z přibližně 80%. Základem programu je obsluha klávesnice, a výpis výsledků nebo chybových hlášek na LCD, a také do terminálu v programu QDevKit. Program načítá čísla pomocí stisku kláves na klávesnici do té doby, než uživatel zmáčkne tlačítko A nebo B, které spustí vyhodnocování zadaného čísla jedním ze dvou implementovaných algoritmů, viz kapitola 2. Samotné algoritmy byly implementovány na základě informací ze zdrojů [1] a [2].

Důležitý je také způsob průběžného ukládání čísla ze vstupu. Tato funkcionalita je realizována pomocí pole, na jehož index daný počítadlem zadaných znaků se uloží znak právě stisknuté klávesy. Jakmile je vstup potvrzen stiskem tlačítka A nebo B, pole čísel (respektive znaků) se převede na číslo a spustí se výpočet. Následně je výsledek převeden zpět na posloupnost znaků a vytisknut na LCD a do terminálu.

4.1 Newtonova (Babylonská) metoda

Newtonova metoda je iterační algoritmus, ve kterém opakujeme výpočet tak dlouho, dokud nedosáhneme požadované přesnosti. Algoritmus konverguje ke správnému řešení v průměru po 7 iteracích. Záleží však na požadované přesnosti. Pro účely tohoto projektu považujeme výsledek za přesný, jestliže je chyba menší než 0,001.

Vzorec, ze kterého je algoritmus odvozen, je následující[1]:

$$a_k = \frac{1}{2} \times \left(\frac{A}{a_{k-1}} + a_{k-1} \right)$$

kde $\boldsymbol{A}$ *je číslo, ze kterého počítáme odmocninu, a* $\boldsymbol{a_n}$ *je výsledek v* $\boldsymbol{k}$*té iteraci*

4.2 Metoda půlení intervalů

Metoda půlení intervalů je také iterační algoritmus. V principu jeden z nejjednodušších a lze jej použít pro výpočet kořenu jakékoli nelineární rovnice. Nejprve si stanovíme dolní a horní mez – tedy interval, na kterém určitě leží řešení dané rovnice. Pro dolní mez volíme hodnotu 0, jelikož výsledek nikdy nebude záporný, pro horní mez pak volíme číslo, ze kterého počítáme odmocninu. Odhad horní meze je možné optimalizovat, ale pro tento algoritmus bychom ušetřili maximálně 2 iterace z celkových 15 (průměrně), takže optimalizaci neprovádíme. Následně se spočítá střed mezi těmito dvěma čísly podle vzorce[2]:

$$a_k = \frac{h_k + d_k}{2}$$

kde $\boldsymbol{a_k}$ *je výsledek,* $\boldsymbol{h_k}$ *je horní mez, a* $\boldsymbol{d_k}$ *je dolní mez, vše v daném kroce* $\boldsymbol{k}$

V dalším kroku vyhodnotíme, jestli je druhá mocnina nalezeného středu větší nebo menší, než číslo, ze kterého počítáme odmocninu. Jinými slovy rozhodujeme, jestli je náš odhad řešení menší nebo větší, než skutečné řešení. Rozhodujeme tímto způsobem:

- Číslo je větší – řešení leží někde mezi dolní mezí, a tímto číslem
- Číslo je menší – řešení leží někde mezi tímto číslem, a horní mezí

Adekvátně tedy upravíme hodnoty horní a dolní meze, a opakujeme výpočet, dokud nedosáhneme chyby menší než 0,001.

4.3 Srovnání

Algoritmus Metody půlení intervalů konverguje ke správnému řešení přibližně 2× pomaleji, než Newtonova metoda při stejné přesnosti, avšak je implementačně nenáročný. Pomaleji v tomto smyslu myslíme, že metoda potřebuje pro výpočet odmocniny v průměru 2× více iterací, a jelikož je algoritmus co se týče výpočtu jedné iterace přibližně stejně výpočetně náročný jako Newtonova metoda, tak i doba výpočtu je přibližně 2× delší.

Newtonova metoda je pro výpočet odmocniny s přesností na 3 desetinná místa zcela přesná, Metoda půlení intervalu nemusí být, i když se nejedná o nepřesnost v pravém slova smyslu. Je

to způsobeno povahou výpočtu (horní a dolní mez). Například pro výpočet odmocniny ze 100 dostáváme po 17 iteracích tyto hodnoty:

Dolním mez: 9,999847

Horní mez: 10,000610

Výsledek: 9,999847

Absolutní rozdíl dolní a horní meze je v tomto případě 0,000763 – chyba je tedy menší, než požadovaná přesnost, přesto je výsledek zdánlivě nepřesný, s chybou 0,001 při výpisu výsledku na 3 desetinná místa.

5 Závěr

Implementované algoritmy nepatří mezi nejrychlejší, ale s ohledem využití aplikace je jejich rychlost dostatečná. Experimentálně bylo změřeno, že výpočet Newtonovou metodou je dvakrát rychlejší, než výpočet pomocí Metody půlení intervalu.

6 Literatura

[1] HONZLOVÁ EXNEROVÁ V. *Výpočet odmocnin od starověku po současnost.* Praha: Univerzita Karlova v Praze, 2014. Matematicko-fyzikání fakulta. Konzultant závěrečné práce Mgr. Zdeněk Halas, Dis., Ph.D.

[2] FAJMON B., HLAVIČKOVÁ I., NOVÁK N. a VÍTOVEC J. *Numerická matematika a pravděpodobnost.* Brno: Vysoké učení technické v Brně, 2014. Fakulta elektrotechniky a komunikačních technologií.