

PLOTTER MERKUR ALFI CONTROLLED BY FITKIT PLATFORM

Ondřej Klubal, Petr Šimon

Master Degree Programme (2), FIT BUT

E-mail: xkluba00@stud.fit.vutbr.cz, xsimon10@stud.fit.vutbr.cz

Supervised by: Aleš Marvan

E-mail: imarvan@fit.vutbr.cz

Abstract: This paper describes a design of a plotter controller using FITkit platform. Driver application does preprocessing of Scalable Vector Graphic files and converts them to commands, which are sent to FITkit. Controller then do rasterization of lines and cubic Bézier curves and drives stepper motors of head and paper feed in both directions.

Keywords: Alfi, Bézier curve, FITkit, Merkur, Plotter, SVG

1 ÚVOD

Cílem projektu bylo vytvoření řídicího systému pro stavebnici souřadnicového zapisovače Merkur Alfi [1]. Ten kresbu vytváří pomocí dvou nezávislých pohybů - pohybem papíru (osa Y) a pohybem pera (osa X), které je umístěno na vozíku tak, že musí být zabezpečený zdvih a spouštění pro vytvoření kresby a pro její přerušení.

Součástí stavebnice je i plošný spoj, na kterém jsou výkonové tranzistory, které spínají cívky krokových motorů a ovládají zdvih pera. Naším úkolem tedy bylo navrhnout a zkonstruovat řídicí systém, který by umožňoval kreslení na tomto plotteru. K řízení jsme se rozhodli využít platformu FITkit. Dali jsme si za cíl pomocí tohoto plotteru vykreslovat vektorové obrázky ve formátu SVG. Z toho důvodu je řešení rozděleno na dvě části:

- Počítačová aplikace - konverze a předzpracování SVG
- Firmware pro FITkit - rasterizace a ovládání plotteru

2 POČÍTAČOVÁ APLIKACE

Konzolová aplikace *SVGPrinter* je vytvořena ve vývojovém prostředí Lazarus. Je tak použitelná jak na operačním systému Windows, tak na operačním systému Linux.

2.1 FORMÁT SVG

Formát SVG [4] je vektorový datový formát založený na XML. Protože se jedná o poměrně komplexní formát, nejsou aplikací podporovány všechny jeho možnosti. Jako zdroj vstupních obrázků jsme uvažovali soubory vytvořené programem na vektorizaci bitmapových obrázků *cr2v*. Výsledné SVG soubory se vyznačují tím, že využívají pouze elementů `<path>` s omezenou množinou příkazů na podmnožinu M, L, C a Z:

- M *x,y* - move to *x,y* - bez kreslení přesune pero na pozici *x,y*.
- L *x,y* - line to *x,y* - nakreslí čáru z aktuální pozice do cílové pozice *x,y*.

- C x_1, y_1 x_2, y_2 x, y - curve to x, y - nakreslí kubickou Beziérovu křivku z aktuální pozice pera. Tento druh křivky je mimo počátečního a cílového bodu definován dvěma pomocnými body.
- Z - closepath - uzavření cesty. Příkaz malé z je pouze synonymem tohoto příkazu.

2.2 TRANSFORMACE SOUŘADNIC

Začneme s přemapováním vstupních souřadnic do pracovního prostoru plotteru. Ten je omezen na 1024 kroků na ose X a 1420 kroků na ose Y . Vertikální a horizontální rozlišení plotteru však není shodné a proto je třeba provést korekci souřadnic, aby nedošlo k porušení poměru stran. 100 kroků na ose X odpovídá přibližně 111,2 krokům na ose Y .

Uživateli je dána možnost změnit velikost obrázku. Ten je vždy programem přepočítán tak, aby se vešel do pracovního prostoru plotteru.

2.3 ODESÍLÁNÍ DAT DO FITKITU

Pro jednodušší zpracování dat na straně kitu jsme zvolili formát odesílaných dat jako pole 16 bitových hodnot. Rozhodli jsme se ponechat příkazy `<path>` i jejich strukturu. Protože u většiny složitých obrázků nelze poslat najednou všechna data kvůli omezené paměti FITkitu, je nutné data posílat po částech. Tento problém řeší řídicí příkazy N a E. Příkaz N je pro FITkit signál, že si má po vykreslení aktuální části zažádat o další. Příkaz E naopak slouží k ukončení tisku. Před tím je však vložena sekvence příkazů, která zaparkuje psací hlavu na výchozí souřadnice.

3 ŘÍDÍCÍ APLIKACE VE FITKIT

3.1 ŘÍZENÍ KROKOVÝCH MOTORŮ PLATFORMOU FITKIT

Pro řízení posuvu hlavy a papíru jsme se rozhodli použít osmitaktní řízení krokových motorů s magnetizací jedné nebo dvou fází. Tento způsob má totiž nejjemnější krok a tedy i nejlepší rozlišení výsledného obrázku.

3.2 KRESLENÍ KŘIVEK

I přes to, že plotter je svým způsobem kreslení určen především pro kreslení vektorové grafiky, je třeba převést vektorový popis obrázku na rastrový, aby bylo možné řídit krokové motory. Naše implementace podporuje kreslení dvou typů grafických primitiv. Jsou to úsečky a Beziérovu kubiky.

Rasterizace [2] úseček se provádí pomocí Bresenhamova algoritmu, který pro výpočet využívá pouze celočíselnou aritmetiku. Je tedy velice vhodný pro rychlé zpracování na MCU.

3.3 BEZIÉROVY KUBIKY

Kubické beziérovu křivky [3] mají velký význam pro praxi, protože jsou skládány z menších dílů, kde můžeme jednoduše definovat spojitost v řídicích bodech jednotlivých segmentů. Beziérovu kubika je určena pomocí dvou krajních a dvou řídicích bodů, které určují tvar křivky. Tyto 2 body ve většině případů neleží na této křivce a jejich vzdálenost může být větší než vzdálenost krajních bodů. Křivka je popsána touto rovnicí:

$$C(t) = \sum_{i=0}^3 \binom{3}{i} t^i (1-t)^{3-i} P_i \quad (1)$$

Rasterizace Beziérovu křivky se často provádí adaptivním algoritmem de Casteljau. Jeho obrovskou nevýhodou je, že pracuje s plovoucí řádovou čárkou (neboť parametr t nabývá hodnot v intervalu $<0, 1 >$) a navíc pracuje na principu dělení křivky na podčásti pomocí rekurze.

Pro rychlejší zpracování na MCU bylo nutné vymyslet jiný způsob rasterizace, protože MCU neobsahuje FP jednotku a její softwarová emulace by byla neúnosně pomalá. Úpravou parametru t , aby místo hodnot z intervalu $\langle 0, 1 \rangle$ mohl nabývat pouze celočíselných hodnot z intervalu $\langle 0, 1024 \rangle$, se zbavíme celočíselné aritmetiky. Celá křivka má tedy rozlišení maximálně 1024 bodů. Pro malé křivky to nebude problém. Několik hodnot parametru t se namapuje na jeden pixel. Pro větší křivky už však nemusí být mezi body dané parametrem t a $t + 1$ vzdálenost jeden pixel, ale větší. V takovém případě se dráha mezi body aproximuje úsečkou.

3.4 HEURISTICKÁ FUNKCE PRO BEZIÉROVY KUBIKY

Při praktickém použití jsme zjistili, že se v SVG souboru většinou používají pouze malé křivky, kde není potřeba přesnost 1024 bodů, ale mnohem méně. V takovém případě pak totiž docházelo ke značnému zpomalování kreslení. MCU muselo spočítat celých 1024 bodů křivky, i když se křivka skládala z menšího počtu bodů. Tento problém jsme vyřešili heuristickou funkcí, která odhaduje, kolik bodů bude nutné spočítat.

Vhodný se ukázal následující vztah, který popisuje vzdálenost mezi body S, P_1, P_2, C :

$$n = \sqrt{(x_s - x_{p1})^2 + (y_s - y_{p1})^2} + \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2} + \sqrt{(x_{p2} - x_c)^2 + (y_{p2} - y_{pc})^2} \quad (2)$$

Protože výpočet odmocnin by byl na platformě FITkit časově velmi náročný, rozhodli jsme se tuto heuristiku předpočítat ještě před odesláním do kitu. Předání hodnot se pak provádí přidáním dalšího parametru příkazu C.

4 ZÁVĚR

Plotter jsme prakticky vyzkoušeli a vykreslili jím mnoho zkušebních obrázků.

Největším problémem se ukázala být hardwarová konstrukce plotteru. Z počátku často docházelo k prokluzování papíru či hlavy s perem. Plotter jsme proto značně upravili a vylepšili, díky čemuž se nám téměř zcela povedlo tyto problémy odstranit. Plotter v akci si můžete prohlédnout na následující adrese: http://www.stud.fit.vutbr.cz/~xsimon10/plotter/angry_wizard.avi. Toto video je také prezentováno na fakultě informačních technologií.

PODĚKOVÁNÍ

Tato práce vznikla částečně za podpory výzkumného záměru MSM0021630528.

REFERENCE

- [1] Doval V.: Souřadnicový zapisovač ALFI [online]. Prešov, 2011-06-07 [cit. 2012-02-25].
Dostupné z WWW: http://www.merkurtoys.cz/download.php?group=produkty2_soubory&id=28
- [2] Geometrie/Rasterizace. In *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, [cit. 2012-02-25].
Dostupné z WWW: <http://cs.wikibooks.org/wiki/Geometrie/Rasterizace>
- [3] Bézierova křivka. In *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, [cit. 2011-12-6].
Dostupné z WWW: http://cs.wikipedia.org/wiki/B%C3%A9zierova_k%C5%99ivka
- [4] The World Wide Web Consortium: Scalable Vector Graphics (SVG) 1.1: Paths [online], Second Edition, 2011-07-16 [cit. 2012-02-25].
Dostupné z WWW: <http://www.w3.org/TR/SVG/paths.html>