

Dokumentácia projektu

**Riadenie modelov pomocou FITkitu -**

**Zariadenie typu plotter pre rezanie a kreslenie**

Vypracovaný v rámci predmetu

Mikroprocesorové a vstavané systémy

|  |  |  |
| --- | --- | --- |
|  | Autor:  Login:  Email: | Ivan Ševčík  xsevci50  xsevci50@stud.fit.vutbr.cz |

Obsah

[Úvod 2](#_Toc404569839)

[Návrh 2](#_Toc404569840)

[Implementácia 2](#_Toc404569841)

[Inicializácia programu 2](#_Toc404569842)

[Hlavná programová slučka 2](#_Toc404569843)

[Diskretizácia plochy a prepočet súradníc 3](#_Toc404569844)

[Ovládanie motora 3](#_Toc404569845)

[Užívateľský vstup 3](#_Toc404569846)

[Zoznam príkazov 4](#_Toc404569847)

[Kresliace kontexty 4](#_Toc404569848)

[Záver 4](#_Toc404569849)

[Prílohy 5](#_Toc404569850)

# Úvod

Zadaním je vytvoriť ľubovoľný model, ktorý bude poskytovať dostatočnú množinu operácií, a následne ho riadiť pomocou FITkitu. Vyžaduje sa možnosť interaktívneho ovládania pomocou terminálu aplikácie QDevKit ako aj autonómneho vykonania zložitejšej operácie.

Ako model, ktorý bude zhotovený a ovládaný, bolo zvolené zariadenie typu plotter, ktoré pozostáva z hlavy pohybujúcej sa v dvoch osiach a funkcionality hlavy, ktorá môže byť rôzna – pri použití laserovej diódy sa môže jednať o rezací nástroj alebo namontovaním pera sa jedná o vykresľovacie zariadenie. Algoritmy pritom ostávajú z veľkej časti rovnaké a mení sa iba hardware.

# Návrh

Po hardwarovej stránke bude zariadenie koncipované z podložky formátu aspoň A4, na ktorej bude umiestnený obrábaný materiál – papier, plošný spoj, drevená doska, atď. Nad touto plochou sa bude v osi Y pohybovať celá mechanika pre pohyb hlavy v osi X, na ktorej už bude umiestnená samotná hlava s nástrojom. Táto mechanika pochádza zo staršej grafickej jednotky XY 4150. Prvý návrh bude ako nástroj používať pero, čiže sa bude jednať o vykresľovacie zariadenie. Pohyb v osiach budú zabezpečovať krokové motory riadené FITkitom a napájané externým zdrojom. Aby sa zamedzilo poškodeniu motorov, každá osa bude obmedzená tlačidlovými spínačmi na začiatku a konci, ktoré budú vymedzovať pracovnú plochu.

Software bude tvorený jedným programom písaným v jazyku C, ktorý sa nahrá do FITkitu a následne bude riadený z terminálu programu QDevKit. Užívateľské príkazy budú predstavovať množinu tvarov zadávaných vo vektorovom formáte a sadu náročnejších operácií. Abstrahovaním od hardware je možné uvažovať o programe ako nástroji pre rasterizáciu vektorových tvarov. Jednotlivé kroky motora totiž rozdeľujú plochu na diskrétne body, medzi ktorými sa hlava presúva. Veľkosť kroku motora potom určuje rozlíšenie. Tým pádom je možné prevziať veľkú časť z kódu projektov do predmetu **Základy počítačovej grafiky**, kde boli prezentované výpočetne efektívne rasterizačné algoritmy. Ako základ boli zvolené celočíselné algoritmy pre rasterizáciu úsečky (Bresenhamov algoritmus) a kružnice, pomocou ktorých je neskôr možné implementovať aj zložitejšie tvary, napríklad Bézierove krivky.

# Implementácia

## Inicializácia programu

Po spustení sa inicializujú jednotlivé piny na portoch podľa potreby na vstup alebo výstup a vykresľovacia hlava sa presunie na začiatok plochy, čím zadefinuje počiatočné súradnice. Následne je program prechádza do hlavnej programovej slučky, v ktorej beží až do odpojenia alebo zastavenia cez uzavretie terminálu QDevKit.

## Hlavná programová slučka

Telo programu predstavuje hlavná programová slučka, ktorá periodicky vykonáva pohyb motorov, prepočítanie algoritmov a spracovanie príkazov od užívateľa. Na konci slučky je vždy uspatie na určitú dobu, ktorá sa odvíja od pracovnej frekvencie krokových motorov. Nižšie hodnoty čakania umožňujú rýchlejšie vykresľovanie, no pri príliš rýchlej frekvencii motor fyzicky nestihne otočiť rotorom a začne preskakovať. Treba sa taktiež vyhnúť rezonančnej frekvencii stroja, ktorá výrazne zhoršuje kvalitu výsledku vibráciami.

## Diskretizácia plochy a prepočet súradníc

Pracovná plocha je prirodzene rozdelená najmenším krokom motorov na dvojrozmernú množinu bodov. Problémom je však nerovnaká veľkosť kroku v X a Y osi a tiež zadávanie súradníc v krokoch motoru nie je užívateľsky prívetivé. Z týchto dôvodov existuje súčasne viacero súradnicových systémov, ktoré sa používajú na rôznych úrovniach programu, a medzi ktorými dochádza k prepočtom. Vstupný súradnicový systém je popísaný nižšie v sekcii Užívateľský vstup. Ten je následne prepočítaný do interného súradnicového systému, ktorý predpokladá body rozmiestnené ekvidistantne (teda v rovnakých rozstupoch) v oboch osiach. Tento systém využívajú rasterizačné algoritmy pre svoje výpočty. Posledný, reálny, súradnicový systém predstavuje pôvodné rozdelenie plochy na kroky motorov, v ktorom môžu mať body rôzne rozstupy v jednotlivých osiach. Interné súradnice sa prepočítavajú do tohto systému po každom kroku rasterizačného algoritmu a ak došlo k zmene v rámci reálnych súradníc, motor spraví krok. Platí pritom, že motor môže v jednom kroku algoritmu spraviť iba jeden krok rotora, preto je maximálna vzdialenosť medzi bodmi interného súradnicového systému daná ako internal\_step\_max = min(motor\_x\_step, motor\_y\_step). Spodná hranica je daná iba použitým dátovým typom, no nemá zmysel používať vysoké interné rozlíšenie ak limitujúci faktor je reálne rozlíšenie, a preto sa ako veľkosť interného kroku volí práve maximum vypočítané podľa uvedeného vzorca.

Konštanty motor\_x\_step, motor\_y\_step sú experimentálne namerané hodnoty reprezentujúce vzdialenosť posunu hlavy v milimetroch pre danú os po vykonaní jedného kroku motora. Sú vyjadrené desatinným číslom, ale keďže algoritmy používajú celočíselnú aritmetiku, sú pri štarte programu prepočítané na celé čísla. Podobne je definovaná aj konštanta internal\_step\_max. V praxi potom konštanta 0.1 znamená, že každý milimeter sa rozdelí medzi 10 celočíselných hodnôt – desatiny milimetra, a tie budú predstavovať najvyššie rozlíšenie.

## Ovládanie motora

Ovládanie motora, teda vykonanie kroku určitým smerom, spočíva v zaslaní správneho slova na výstupný port. Zopnutím pinov sa zopnú cievky v motore a korektnou postupnosťou zapínania a vypínania týchto cievok sa docieli otáčanie rotora. Preto je v programe uložená postupnosť slov do poľa v poradí, v akom majú po sebe nasledovať, a pre každý motor existuje index do tohto poľa. Po použití posledného slova sa index vracia na začiatok poľa riadiacich slov.

Motory vyžadujú napätie 24 voltov, teda nemôžu byť napájané priamo FITkitom, ktorý poskytuje len napätie používané v logických obvodoch, čo je bežne 5 voltov. Aj pri nižších napäťových požiadavkách by však bola stále problémom prúdová záťaž, pretože piny sú priamo napojené na výstupy mikroprocesoru a tam sa hodnoty pohybujú rádovo v miliampéroch. Z tohto dôvodu bol zostrojený predzosilňovač, ktorý prevádza logické úrovne na potrebných 24 voltov, a je napájaný externým zdrojom z bežnej elektrickej siete. Schéma zosilňovača je uvedená v prílohách.

## Užívateľský vstup

Užívateľ komunikuje s FITkitom a riadi stroj pomocou množiny príkazov zadávanej do terminálu aplikácie QDevKit. Nasledujúci zoznam popisuje ich použitie a význam. Všetky zadávané súradnice predstavujú body na ploche, pričom počiatok je v ľavom hornom rohu plochy a súradnice rastú smerom dole a doprava. Ako jednotka bol použitý milimeter, súradnica <5; 30> teda predstavuje bod 5 milimetrov vpravo a 3 centimetre nadol od počiatku. Program akceptuje aj záporné hodnoty, no ich využitie vzhľadom na umiestnenie počiatku je minimálne.

Zoznam príkazov

**LINE** x1 y1 x2 y2 Vykreslí priamku z bodu <x1; y1> po bod <x2; y2>, pričom pri presune do počiatočného bodu zdvihne pero, teda preruší kreslenie.

**CIRCLE** x1 y1 R Vykreslí kružnicu so stredom v bode <x1; y1> a polomerom R, pričom pri presune do vrcholu kružnice zdvihne pero, teda preruší kreslenie.

**CUT** x1 y1 Pokračuje v kreslení priamky z bodu, na ktorom sa hlava aktuálne nachádza, do bodu <x1; y1>.

**DEMO** Vykreslí demo z pamäte mikroprocesoru, ktoré bolo uložené pri kompilácii programu. Nachádza sa v súbore demo.h a umožňuje jednoducho napísať komplexnú postupnosť príkazov, ktoré sa majú vykonať. Syntax a sémantika je veľmi podobná ako u vyššie uvedených príkazov, iba navyše odpovedá syntaxi jazyka C.

V budúcnosti sa navyše plánuje podpora pre nahrávanie súborov vo formáte HPGL, obsahujúcich kompatibilné príkazy, ktoré budú transformované do podporovanej množiny príkazov. Z dôvodu komplexnosti komunikácie s FITkitom a nutnosti vytvorenia samostatnej aplikácie tento spôsob komunikácie nebol zatiaľ implementovaný. V budúcnosti sa taktiež predpokladá nahradenie FITkitu jednoduchším mikroprocesorom napríklad rady ATMega a komunikácia potom bude teda prebiehať odlišne, pravdepodobne po sériovej linke USART.

## Kresliace kontexty

Pre zachovanie informácií o stave vykresľovania daného tvaru medzi jednotlivými krokmi sa používajú kontexty. Kontext je predaný ako parameter funkcii, ktorá vykoná jeden krok vykresľovania a nastaví premenné pre ďalší krok. Po prijatí príkazu sa kresliaci kontext inicializuje potrebnými hodnotami a uloží sa do vyhradenej stavovej premennej typu union, ktorá umožňuje prekryť pamäťové miesta pre rôzne typy kontextov a šetriť tak cennou pamäťou mikroprocesoru.

# Záver

Výsledkom projektu je pracujúce zariadenie, ktoré spĺňa formálne požiadavky na projekt. V budúcnosti sa plánuje využívanie zariadenia, čomu však budú predchádzať isté úpravy. V prvom rade to bude výmena riadiacej jednotky, teda FITkitu, za jednoduchší a sériovo vyrábaný mikroprocesor. Dôvodom je nutnosť navrátenia FITkitu po istej dobe a tiež potreba opätovného prepájania. V prípade mikroprocesora bude možné vytvoriť trvalé spoje, čím sa zapojenie zjednoduší. Program písaný v C je však dobre prenositeľný, a preto sa nepredpokladajú žiadne komplikácie. Ďalším krokom potom bude napísanie riadiaceho programu na strane PC, ktorý nahradí QDevKit. Na komunikáciu sa využije rozhranie USART, pre ktoré existujú lacné a dostupné USB adaptéry.

# Prílohy