



Mikroprocesorové a vestavěné systémy (IMP)
Řízení modelů pomocí mikrokontroleru

v Brně
31. prosince 2017

Václav Martinka
xmarti76

Obsah

1	Zadání	1
1.1	Konkrétní řešení	1
2	Návrh řešení	1
2.1	Spínání ventilátoru	1
2.2	Snímání stavu ventilátoru	2
2.3	Vstup od uživatele	2
3	Implementace	2
3.1	Kontrola ventilátorů	2
3.2	Vykonání programové sekvence	2
4	Popis chování	3
4.1	Programové sekvence	3
4.2	Výstup na LCD	3
5	Závěr	3

1 Zadání

Zvolte vhodný model (hotový, LEGO, Merkur nebo jiná stavebnice), který bude disponovat alespoň třemi různými funkcemi/pohyby funkčních bloků (např. bagr, jeřáb, robot apod.) Předpokládám, že zapojíte svoji tvořivost, aby výsledkem byl smysluplně vyhlížející a fungující celek.

Navrhnete vhodné rozhraní pro připojení pohybových elementů s FITkitem a vytvořte ovládací program v jazyku C pro MSP430, pomocí kterého bude možné interaktivně řídit všechny úkony modelu (pomocí příkazů zadávaných přes terminál s vhodně nastavenými parametry, případně prostřednictvím klávesnice FITkitu). Dále navrhnete alespoň dvě programové sekvence, na kterých budete demonstrovat činnost modelu realizujícího posloupnost zvolených povelů (jediným příkazem model vykoná složitější úkon).

1.1 Konkrétní řešení

Původně jsem měl v plánu ovládat plotr sestavený ze stavebnice merkur. Bohužel zde nastaly konstrukční problémy. Proto jsem se uchýlil k náhradnímu řešení - ovládání ventilátorů. Tedy možnosti jednotlivé ventilátory zapnout nebo vypnout včetně kontroly jejich funkčnosti.

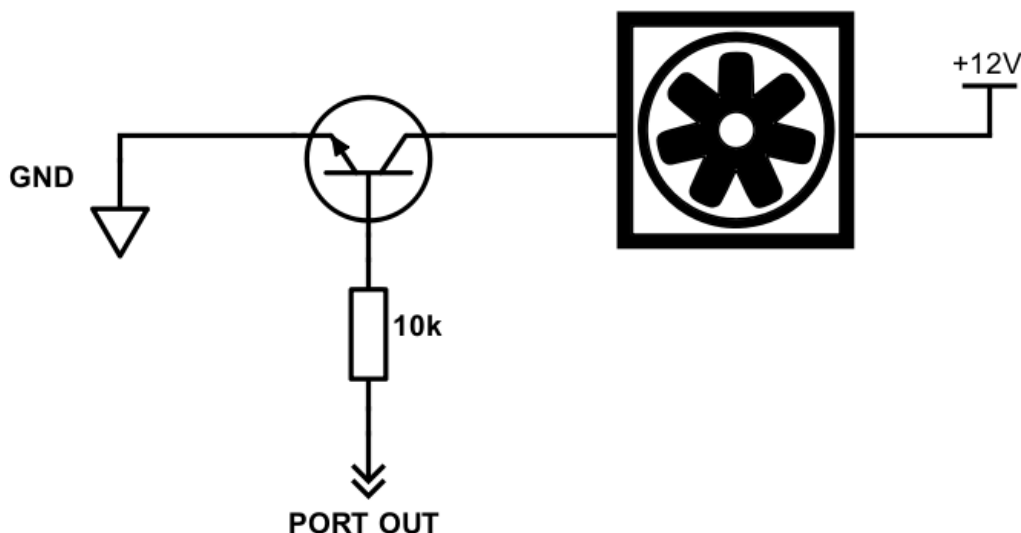
Využití tohoto systému si dokážu představit kdekoli, kde je potřeba chladit nějaké zařízení. Zejména v případě, že by se vstupy z klávesnice nahradili (nebo doplnili) vstupy z teplotních čidel.

2 Návrh řešení

Zadání obsahovalo několik bloků, které bylo nutné vyřešit. Ty jsou na sobě implementačně nezávislé, tudíž bylo možné je řešit postupně.

2.1 Spínání ventilátoru

Jelikož MCU není schopný na pinu dodat dostatečně vysoké napětí (12 V) ani proud (0,2 A) je nutné zapojení doplnit o spínací NPN tranzistor.



Obrázek 1: Schéma zapojení ventilátoru k MCU

2.2 Snímání stavu ventilátoru

Pro projekt jsem využil 3vodičové ventilátory. Tedy takové, které kromě napájení obsahují ještě výstup z Hallovy sondy.¹ Na tomto vodiči se střídavě objevuje logická 0 a 1, podle toho, o kolik je otočena vrtule ventilátoru. Některé ventilátory vykazovali problémy, pokud byl tento vodič připojen přímo na vstup MCU. Bylo proto nutné jej přes rezistor (s hodnotou 10 k Ω) připojit na napájení. Zároveň z důvodu ochrany MCU není vodič připojen přímo na vstup, ale je využit ještě předřadný rezistor.

Signál z tohoto vodiče lze snadno využít pro kontrolu funkce ventilátoru – pokud se hodnota nemění, tak se ventilátor zastavil. Taktéž by bylo možné tento signál využít pro měření otáček ventilátoru.

2.3 Vstup od uživatele

Jako uživatelský vstup jsem si vybral klávesnici a to zejména z důvodu jednoduchosti ovládání. V případě, že by zařízení mělo být ovládáno centrálně z nějakého vzdáleného místa, bylo by jistě vhodnější mít možnost zařízení ovládat i přes terminál.

V reálném systému bych pak zařízení ještě doplnil o teploměry, aby bylo celé automatizované.

3 Implementace

Program pro MCU jsem implementoval v jazyce C. Vše je umístěno v jednom souboru a byli využity pouze standardní knihovny (`stdbool.h`, `ftitkitlib.h`, `keyboard/keyboard.h`, `lcd/display.h`).

Základní inicializace je umístěna v těle funkce `fpga_initialized()`. Ta se postará o nastavení logických vstupů/výstupů. Nastavení časovače a LCD.

Hlavní logika programu je pak umístěna v nekonečné smyčce ve funkci `main()`. Během této smyčky je obsloužena klávesnice, dojde ke kontrole, zda se ventilátory točí a vykreslení aktuálního stavu na LCD.

3.1 Kontrola ventilátorů

Nabízelo se několik řešení, jak tento problém vyřešit. Nakonec jsem se rozhodl pro přímočaré řešení. Načtu s odstupem 1 ms 8krát po sobě hodnotu ze vstupu, všech 8 hodnot si uložím do jednoho bytu. Pokud tento byte obsahuje 0xFF nebo 0x00, tedy samé 0 nebo 1, tak mám téměř jistotu, že se ventilátor netočí. Abych zaručil, že se otáčky ventilátoru neshodují s frekvencí MCU, tak během snímání provádím i jinou činnost (příprava hodnot pro vykreslení na LCD). Díky tomu vím, že hodnoty nejsou snímány pravidelně a je tudíž velmi malá šance, že nastanou falešná hlášení.

3.2 Vykonání programové sekvence

Při každém průchodu hlavní smyčky se volá funkce `running_program()`. Ta je složena z přepínače `switch` s argumentem ID spuštěné sekvence.

Na daném návěští se nachází další `switch`, kde je jakožto argument čítač kroků sekvence. Tento čítač inkrementuji dle potřeby. Takto lze snadno napsat posloupnost jednotlivých kroků včetně pod-
mínek, kdy se přechází na další krok.

¹Hallova sonda je elektronická součástka, jejíž činnost je založena na technickém využití tzv. Hallova jevu. Používá se pro měření a automatickou regulaci magnetických polí, ovládání velkých elektromotorů, bezkontaktní tlačítka, mechanické snímače (poloha, otáčky, zrychlení) apod. [zdroj Wikipedie (https://cs.wikipedia.org/wiki/Hallova_sonda)]

4 Popis chování

Jak už jsem zmínil, zařízení se ovládá pomocí klávesnice:

1 – 9 Tyto klávesy slouží k zapnutí/vypnutí konkrétního ventilátoru. Protože program byl vytvořen jen pro tři ventilátory, jsou funkční pouze klávesy 1,2 a 3.

***** Tato klávesa sepne všechny ventilátory bez ohledu na jejich aktuální stav.

Opak k *, tedy vypnutí všech ventilátorů.

A-D Klávesy určené k sepnutí jednotlivých programových sekvencí. Jsou implementovány pouze 2 sekvence: A a B. Spuštění jedné ze sekvencí uzamkne klávesnici až do jejího vypnutí. Spuštěná sekvence je signalizována pomocí svítivé diody D5.

0 V případě, že je spuštěna nějaká sekvence, tak stisknutím dojde k vypnutí právě spuštěné sekvence, odemknutí klávesnice a zastavení všech ventilátorů.

4.1 Programové sekvence

A V této sekvenci se postupně spouští ventilátory 1, 2 a 3. Vždy po 7 sekundách, přičemž při sepnutí nového ventilátoru se ten předchozí vypne. Sekvence běží ve smyčce až do jejího ukončení.

Využití by bylo v případě, že bychom potřebovali řídit velké (nebo velké množství) ventilátory s vysokým odběrem a přitom by nebyl vyžadován jejich stálý běh. V tom případě by bylo jistě vhodnější vždy spustit jen jeden, než střídavě zapínat/vypínat všechny naráz.

B Tato sekvence se stará o bezpečné chlazení. Na začátku je spuštěn první ventilátor, v případě, že selže, dojde ke spuštění druhého. To se opakuje, dokud neselezou všechny ventilátory.

4.2 Výstup na LCD

Po celou dobu běhu programu je na LCD zobrazován aktuální stav všech ventilátorů. Ten nabývá tří hodnot:

OFF Značí vypnutý ventilátor

OK Ventilátor je zapnutý a nehlásí žádné problémy

ERR Došlo k chybě ventilátoru

5 Závěr

I přes prvotní problémy s plotrem se mi nakonec podařilo vytvořit funkční zařízení. Výsledný kód by měl být poměrně snadno rozšiřitelný na větší množství ventilátorů, což byl jeden z mých cílů.

Co se týče možnosti rozšiřitelnosti do budoucna, tak by to bylo doplnění snímání počtu otáček, řízení rychlosti otáček (to vyžaduje jiné ventilátory) nebo přidání teploměrů.