



IMP PROJEKT - HRA HAD

NIKOLA MACHÁLKOVÁ

[Video](#)

ÚVOD DO PROBLÉMU

Motivace

Výsledkem projektu je hra HAD, ve které se had pohybuje po dvou maticových LED displejích připojených na FITkit 3, kdy displej je o velikosti 8 sloupců a 8 řádků. Hra nemusí obsahovat sbírání ovoce ani zvětšování či zmenšování délky hada – délka tedy zůstává fixní.

Přehled využitého hardware a software

Jako hardware bylo použito zařízení FITkit 3 s maticovými displeji. Maticové displeje jsou o velikosti 8*16 bodů a jsou připojeny k FITkitu pomocí konektoru P1 platformy FITkit 3 a konektoru P3, který je i příslušně označen.

FITkit obsahuje hned několik tlačítek, pro potřeby projektu se však využívala pouze tlačítka s vývody P26, P10, P12 a P27, respektive nahoru, doprava, dolů a doleva. Poslední tlačítko P11 by tedy teoreticky mohlo sloužit k případnému rozšíření – třeba na reset polohy hada, nebo na prodlužování hada, nebo na generaci ovoce ke sbírání.

Jako software byl použit Kinetis Design Studio (KDS). Jelikož projekt pracuje pouze s displeji a FITkitem, nebyla potřeba vytvářet nebo používat další technologie, bohatě stačilo výše zmíněné.

Knihovna se v projektu používala jediná – *MK60DZ10.h*. Dále se deklarovala makra – pro *GPIO* pro přístup k pinům a pro nastavení jednotlivých pinů. Tyto informace jsou převzaty z testovacího programu, získaného u zadání projektu.

POPIS ŘEŠENÍ

Na začátku programu se definují makra a globální proměnné *xsnake*, *ysnake* a *direction*.

Xsnake a *ysnake* jsou pole o velikosti 4 (aneb předem určená délka hada), které obsahují souřadnice x a y jednotlivých bodů/částí těla na displejích. Na začátku jsou zavedeny pouze x-ové souřadnice a had má tedy hlavu na sloupci s indexem 4, nultý řádek.

Proměnná *direction* je pak jeden znak, který určuje směr pohybu hada. Jedná se vždy o začátek slov *up*, *right*, *down*, *left*. Na začátku má pozici *right*, had se tudíž bude pohybovat směrem doprava, dokud není stisknuto jiné tlačítko.

Program poté přechází do *main()*, kde zavolá funkci *SystemConfig()*. Tato funkce nastavuje potřebné MCU zařízení a je převzatá z testovacího programu. Je však rozšířená o nastavení jednotlivých pinů a vymazáním a zapnutím IRQ.

Program by neměl nikdy opustit funkci *main()*, proto je důležité, aby další řádky kódu byly obsaženy v cyklu implementovaném jako *while(1)* pro nekonečný cyklus.

Následuje další cyklus, tentokrát nastavení pozice těla pro body 2 – 4. Každý bod si bere souřadnice x a y bodu před ním, a tím se tedy korektně posunuje. Jakmile se souřadnice nastaví, zjistí se, které tlačítko (pokud nějaké) bylo stisknuto, a kam se bude tedy had, resp. jeho hlava, dále pohybovat. Pokud interrupt přijde, řeší se ve funkci *change_direction()*, v níž se nastavuje proměnná *direction* na příslušný směr.

Program se tedy přesune do funkce *choose_direction()*, ve které se přičítají či odčítají souřadnice podle směru. Hned na začátku se nastaví sloupec, ve kterém je hlava hada. Poté se přejde do switche s proměnnou *direction*, která rozhoduje o způsobu pohybu. Též se zde detekuje přetečení řádku/sloupce. Pokud hlava přesáhne index sloupce 7, vrátí se zpět na 0. Pokud hlava přesáhne index řádku 15, vrátí se zpět na 0.

Jakmile je rozhodnuto o souřadnicích, program se vrátí zpět do funkce *main()* a projde posledním cyklem. Tentokrát se jedná o cyklus, který funguje jako zpoždění, a volá funkci *light()*, která rozsvítí příslušné řádky a sloupce. Funkce *light()* pak nastavuje řádek, na kterém se daný bod nachází pomocí bitových přesunů příslušných k řádkům na displejích, a to pomocí příkazu *PTA->PDOR /= GPIO_PDOR_PDO (1 << x)*, kde x právě odpovídá číslu příslušného řádku. Jedná se tedy o řádky (v pořadí od 0): 29, 27, 7, 28, 25, 9, 24 a 26.

Program se poté vrací zpátky na začátek cyklu *while* a takhle se donekonečna opakuje.

ZHODNOCENÍ

Testování fungovalo jednoduše – po dokončení programování se vyzkoušely veškeré use-case, které by mohly nastat. Zkoušel se tedy přechod z jedné strany na druhou, rychlá změna pohybu i změna směru v rámci pouze sloupce/pouze řádku. Jako další se testovala rychlost, jestli se had nepohybuje moc pomalu, nebo jestli odezva na interrupt netrvá moc dlouho.

Jak lze ale vidět na [videu](#), všechny tyto testy had úspěšně a zdárně zvládl, a může se tedy zkonstatovat, že testování proběhlo úspěšně a program funguje tak, jak má.

VLASTNÍ HODNOCENÍ

$$\Sigma = (K_1 + K_2 * F/5) * (E + F + Q + P + D) = (0.25 + 0.75 * 5 / 5) * (1 + 5 + 3 + 1 + 3) = \mathbf{13\ b.}$$