



Hra HAD na maticovom displeji

Dokumentácia k projektu z predmetu IMP

Autor: Martin Ivančo (xivanc03)

Obsah

ÚVOD	1
POPIS OVLÁDANIA	1
EXTERNÉ SCHÉMY ZAPOJENIA	1
SPÔSOB RIEŠENIA.....	2
FUNKCIA START_GAME()	2
ZAUJÍMAVÉ ČASTI KÓDU.....	2
PREBRATÉ ČASTI KÓDU	3
ZÁVER	3

Úvod

Táto dokumentácia popisuje implementáciu projektu z predmetu Mikroprocesorové a vstavané systémy v akademickom roku 2017/2018. Úlohou projektu bolo vytvoriť hru HAD pre MCU MSP430 zobrazovanú na 8x8 maticovom displeji pripojenom cez I/O rozhranie FITkitu. Aplikácia mala byť prispôsobená obmedzeným rozmerom hracej plochy.

Popis ovládania

Po spustení aplikácie sa hra nainicializuje a zobrazí na displeji. Po stlačení tlačidla s číslom 5 (START) sa had rozhybe a začne sa hra. Had sa pohybuje konštantnou rýchlosťou bez prestávky, úlohou hráča je meniť jeho smer. Ten sa mení tlačidlami s číslami 2 (smer HORE), 4 (smer VĽAVO), 6 (smer VPRAVO) a 8 (smer DOLE). Had sa môže otočiť len o 90 stupňov pri jednom pohybe, takže ak hráč stlačí napríklad tlačidlo DOLE v stave, keď had smeruje HORE, had nezmení smer.

Okrem hada sa na ploche vždy nachádza jedlo. Úlohou hráča je korigovať hada tak, aby k nemu prišiel a mohol ho zjesť. Po zjedení jedla sa had predĺži o jedno políčko a automaticky sa vygeneruje nové jedlo na náhodnej voľnej pozícii na hracej ploche.

Pokiaľ hráč nezvládne ukorigovať hada a narazí hlavou do chvosta, hra sa končí neúspešne. Pokiaľ sa mu však podarí tejto situácii vyhnúť až pokým had nebude mať dĺžku 50 políčok, vyhráva.

Keďže hracia plocha je relatívne malá, hra neobmedzuje hráča hranicami poľa. Pokiaľ had vyjde z hracej plochy, jednoducho postupne vyjde na opačnej protiľahlej strane.

Externé schémy zapojenia

Hra sa zobrazuje na LED maticovom displeji o rozmeroch 8x8, ktorý je k FITkitu pripojený prostredníctvom poľa JP9. Do tohto poľa je zavedených hneď niekoľko portov, avšak pre túto úlohu sa využívajú len tri. Finálne zapojenie môžete vidieť na obrázku č. 1 nižšie.

Port P6 sa využíva v celom rozsahu a slúži na aktiváciu stĺpcov displeja. V poli JP9 je zavedený na piny 31-38.

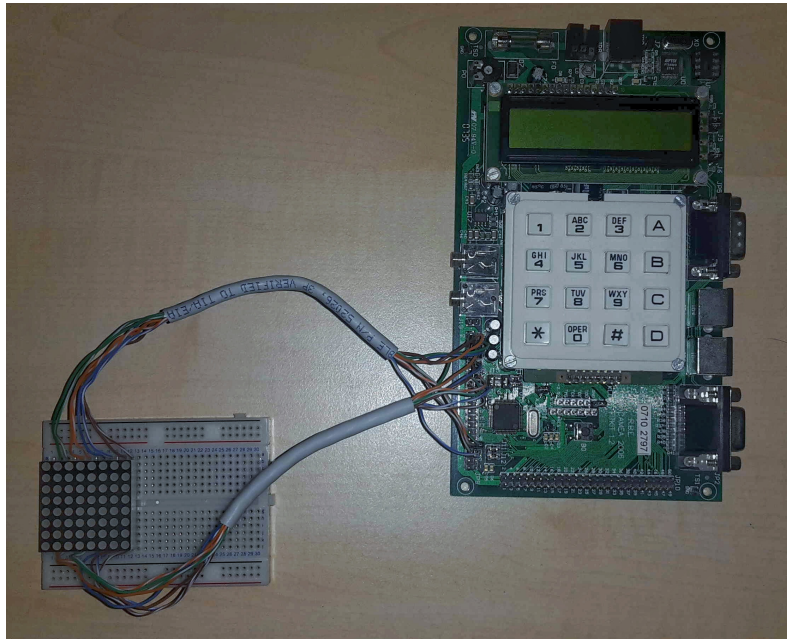
Port P4 sa využíva v rozsahu 0-3, teda spodné 4 bity. Spoločne s portom P1 slúži na aktiváciu riadkov displeja. Port P4, resp. jeho využité bity sú zavedené na piny 5-8. Pomocou tohto portu sa teda aktivujú prvé 4 riadky displeja. Port P1 sa využíva v rozsahu 4-7, teda vrchné 4 bity a slúži na aktiváciu zvyšných 4 riadkov displeja.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	X ₀	Y ₀		X ₁	Y ₁		X ₂	Y ₂		X ₃	Y ₃		X ₄	Y ₄		X ₅	Y ₅		X ₆	Y ₆		X ₇	Y ₇

Využitie pinov maticového displeja

Bit	0	1	2	3	4	5	6	7
P6 (x)	37	38	35	36	33	34	31	32
P4 (y)	7	8	5	6				
P1 (y)					27	28	25	26

Zapojenie využitých I/O portov na piny poľa JP9



Obrázok č.1: Finálne zapojenie displeja k FITkitu

Spôsob riešenia

Po inicializácii a nastavení výstupných bitov program vchádza do hlavnej slučky. V nej vždy nainicializuje hada pomocou funkcie `initialize_snake()` a následne čaká na užívateľa, kým neodštartuje hru. Potom zavolá funkciu `start_game()` ktorá sa stará o priebeh hry a bude popísaná nižšie. Podľa toho či sa užívateľovi podarilo vyhrať alebo prehral sa následne nastaví displej a program čaká kým užívateľ stlačí tlačidlo na reštartovanie hry.

Funkcia `start_game()`

Funkcia `start game` obsahuje slučku z ktorej vychádza jedine ukončením hry víťazstvom alebo prehrou. Najprv sa zavolá funkcia `game_step()` ktorá vykoná krok v hre. Potom sa skontroluje či hráč nevyhral alebo neprehral a ak hra pokračuje tak chvíľu čaká a zobrazuje stav na hraciu plochu.

Krok v hre funguje tak, že najprv sa nájde súradnica pozície práve pred hlavou hada a zistí sa, či je voľná alebo či sa na nej nachádza jedlo alebo telo hada. Podľa situácie sa potom vykoná krok a had sa posunie o jedno políčko ďalej v aktuálnom smere, alebo had zje jedlo a predĺži sa o jedno políčko, alebo funkcia vracia hodnotu 1 ktorá znamená prehru.

Po posunutí hada alebo zjedení políčka sa vždy musí zavolať funkcia `redraw()`, ktorá prepíše stav hracej plochy, uložený v dvojrozmernom poli `display`.

Zaujímavé časti kódu

Z hľadiska maticového displeja je veľmi zaujímavá funkcia `show_display()`, ktorá na displej vykreslí hraciu plochu podľa údajov v poli `display`. Funguje tak, že postupne aktivuje riadky displeja a v nich vždy jednotlivé stĺpce. Taktiež využíva oneskorenie aby sa LEDky stihli úplne rozsvietiť.

Zaujímavý je tiež spôsob uloženia stavu hada. Hlava hada má vždy uloženú aktuálnu pozíciu v premennej `snake_head`. Aktuálny smer je uložený v premennej `snake_direction`. Zvyšok tela hada je uložený pomocou poľa `snake`, ktoré obsahuje smerovacie premenné ktoré určujú pozíciu ďalšej časti tela. Aktuálna dĺžka hada sa nachádza v premennej `snake_size`.

Prebraté časti kódu

Pri programovaní projektu som vychádzal z demo aplikácie na snímanie stlačených kláves na FITkite a následné vypísanie do terminálu. Kód pre MCU teda obsahuje základné funkcie potrebné pre chod aplikácie, avšak všetky sú prispôsobené pre tento projekt. Kód pre FPGA je však prevzatý kompletne. Prebratý a upravený je aj projektový súbor, ktorý obsahuje zoznam použitých knižníc a zdrojových súborov, avšak popis aplikácie je upravený pre tento projekt.

Záver

Projekt bol vyvíjaný a testovaný na FITkite verzii 1.2. Pri testovaní sa vyskytol problém s chybným výstupným poľom FITkitu, avšak po výmene funguje všetko podľa zadania.