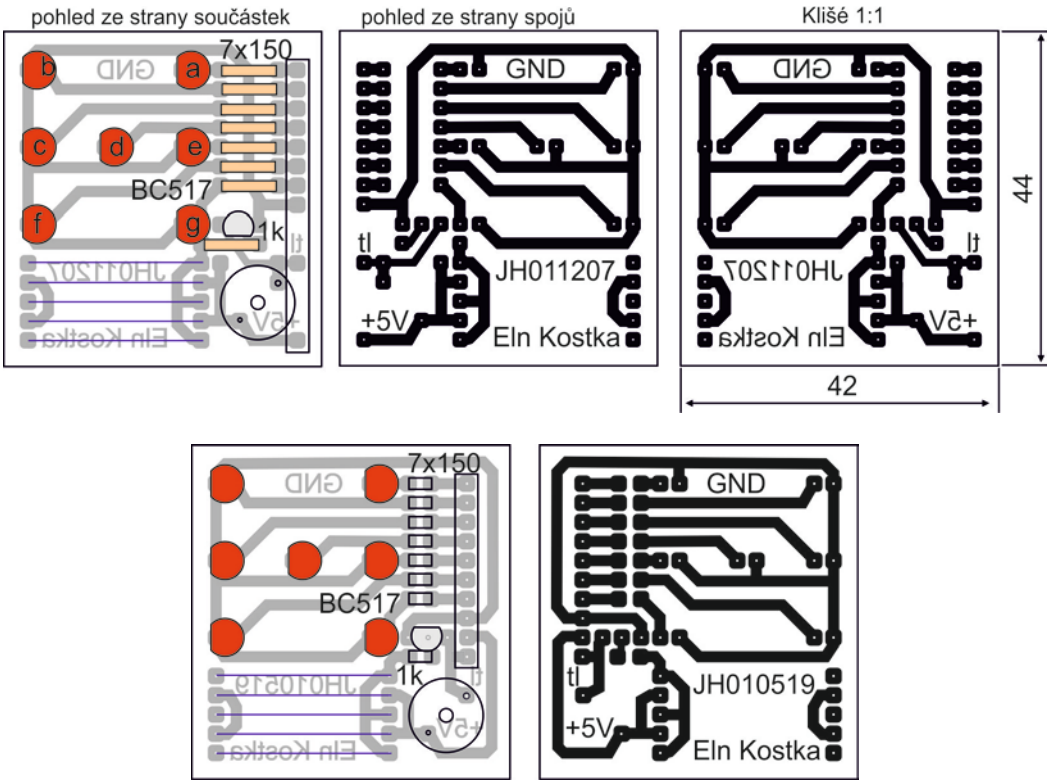
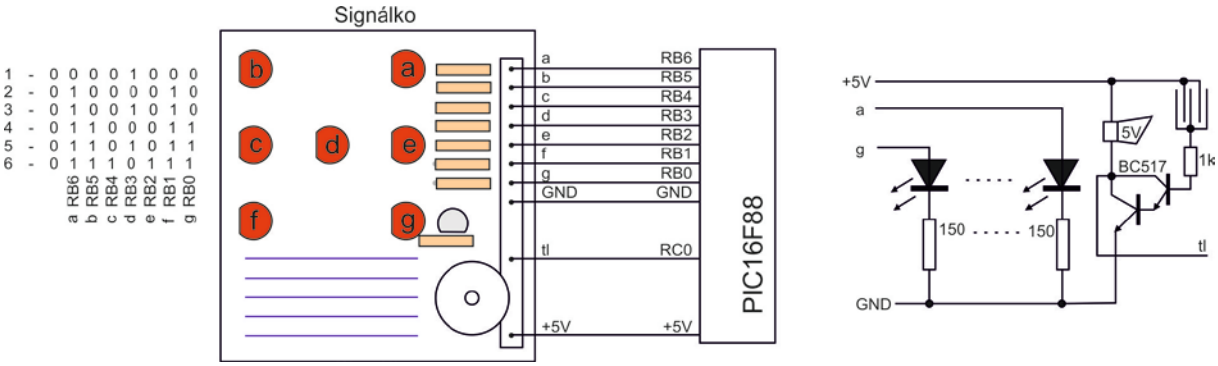
Elektronická kostka

Schéma zapojení elektronické kostky, klišé pro výrobu plošného spoje a možnost použití běžných odporů, nebo odporů SMD řady 1206



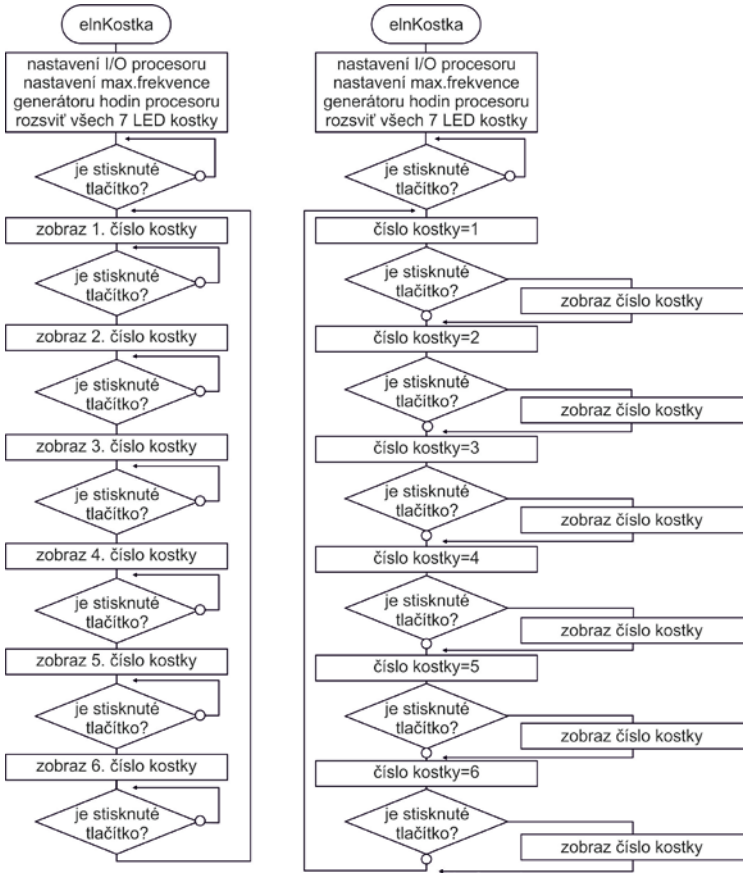
Propojení elektronické hrací kostky s mikrokontrolérem PIC16F88 (signálové schéma)



Ze schématu vyplývá, že vysoká úroveň rozsvěcuje LED a nízká ji zhasíná. Stisknuté tlačítko (dotyk bezkontaktního) rozhouká sirénku a vystaví se na výstup tl logická úroveň L. Není-li tlačítko stisknuté (nedotýkáme se jej), je na výstupu tlačítka úroveň H.

Zadání: Naprogramujte elektronickou kostku. Po zapnutí se rozsvítí všech 7 Led. Po stisku tlačítka se odstartuje losování. Postupně se bodou zobrazovat čísla maximální rychlostí. Po uvolnění tlačítka bude zobrazeno číslo, které svítilo v okamžiku uvolnění tlačítka. Aby bylo losování co nejrychlejší a nebylo tedy možné odhadnout, kdy pustit tlačítko, nastavíme procesoru maximální velikost kmitočtu generátoru hodin procesoru.

Budeme-li předpokládat, že délka stisku tlačítka bude, díky vysoké rychlosti střídání čísel pokaždé jinak dlouhá, můžeme jednoduše zobrazovat čísla třeba v pořadí od 1 do 6. Hrozí-li nebezpečí, že by se mohly velmi krátké stisky jen velmi málo lišit délkou stisku od sebe, bylo by lepší zobrazovat čísla v náhodném pořadí, například 5, 3, 6, 4, 2, 1. Ideální je, aby pokaždé začínala od náhodného čísla a délka stisku vnášela do losování další náhodnou veličinu. Podívejme se na obě alternativy z pohledu ukecaného vývojového diagramu.

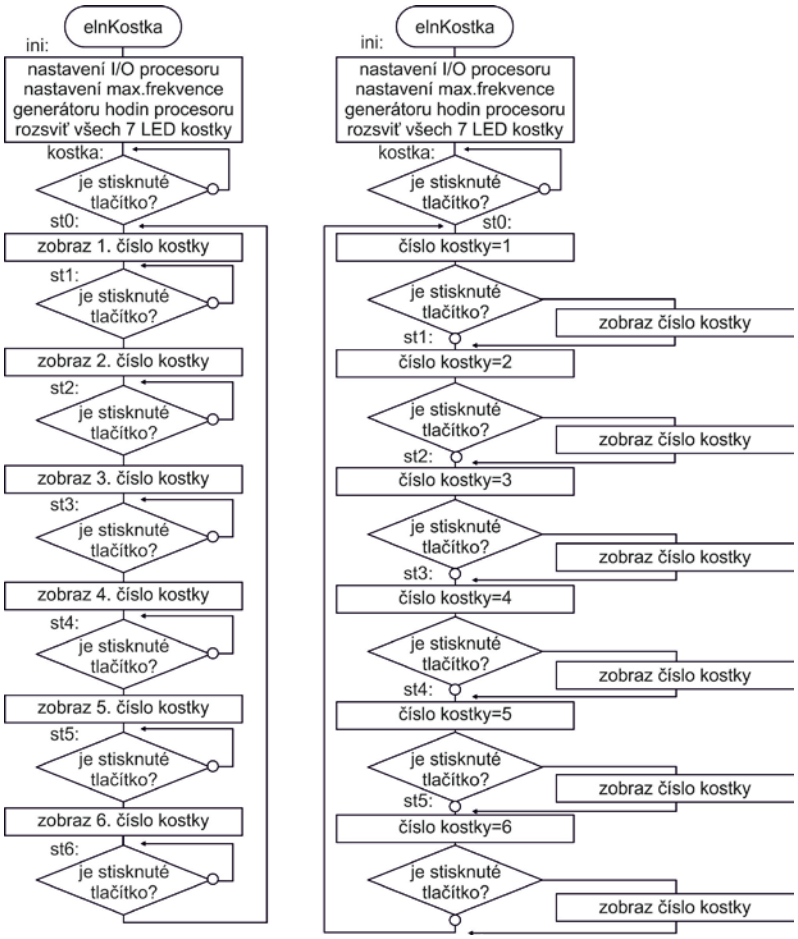


Prostudujte si oba algoritmy a pokuste se definovat podrobně jejich činnost a popsat zásadní rozdíl mezi nimi.

Činnost	PICAXE20M2 (basic)	PIC 16F88 (asm)
Konfigurace I/O procesoru PORTA jako vstupy PORTB jako vystupy	dirsC = 0 dirsB = 255	bsf STATUS,RP0 movlw 0FFh movwf PORTA movlw 00h movwf PORTB movwf ANSEL
generátor hodin procesoru nastav na maximální frekvenci	setfreq M32 ;na 32 MHz	movlw 70h ;na 8MHz movwf OSCCON bcf STATUS,RP0
zobraz 1. číslo kostky a čekání na stisk tlačítka (levý vývojový diagram)	pinsB=8 st1: if pinC.0=1 then goto st1:	movlw 08h ;zobrazení čísla 1 movwf PORTB st1: btfsc PORTA,0 goto st1
zobraz 1. číslo kostky a čekání na stisk tlačítka (pravý vývojový diagram)	cisloKostky=8 if pinC.0=1 then goto st1 pinsB=cisloKostky ;zobraz cislo kostky st1:	movlw 08h movwf cisloKostky btfsc PORTA,0 goto st1 movfw cisloKostky movwf PORTB st1:

V tabulce můžeme vidět, jaký rozdíl je v programu v Basicu PICAXE a Assembleru mikrokontrolérů PIC. Rozdíl není nikterak významný a zdá se, že assembler vyžaduje mnohem víc instrukcí a proto bude pomalejší. Opak je pravdou. I když procesor PICAXE20M2 poběží na kmitočtu 32 MHz, provádění jednotlivch instrukcí trvá desetiny až jednotky ms (tedy max. 10 tisíc instrukcí za sekundu). Pokud poběží procesor PIC16F88 na 8 MHz (vnitřní RC oscilátor), bude trvat jedna instrukce pouze 0.5 μs (2 miliony instrukcí za sec). Pokud bychom u něj použili vnější generátor hodin 20 MHz, pak by každá instrukce trvala pouze 0.2 μs (pět milionů instrukcí za sekundu).

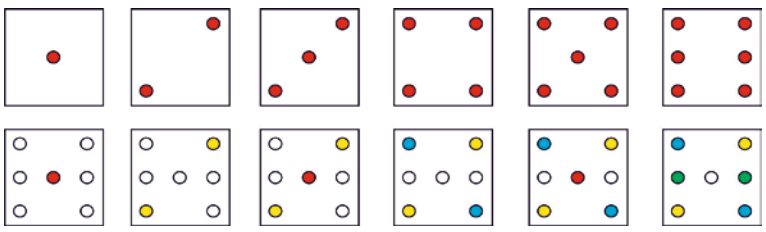
Do vývojových diagramů doplníme z programů návěští (labels) kam vedou skoky v programu.



To nám významně usnadní orientaci v programu jak při simulaci, tak i při orientaci v již hotovém programu pro vyhledávání funkčních celků programu, nebo i jeho budoucí modifikaci.

Pokud vytváříme složitější systgémy, není od věci je navrhovat a rozcházet postupně. Tím předejdeme vzniku vícenásobných chyb, které se velmi těžko hledají a odstraňují.

V našem případě může být takovým problematickým místem zakódování čísel kostky.



Pro ověření správného řízení zobrazování čísel je vhodné před tím, než budeme čísla rychle generovat, je generovat pomalu, abychom si byli jisti jejich správným zobrazováním.

Pokud funguje bez problémů zobrazování, pak již nic nestojí v cestě odstranit čekání a doplnit reakce na tlačítka.

Možná modifikace: Vzhledem k počtu 1 až 6 kamenů na kostce jsme zvolili 7 výsupů pro jejich zpobrazení. Zamysleme se, zada není možné tento počet redukovat. Označme si barevně body, které svítí současně. Zjistíme, že 7 výstupů se dá redukovat na 4. Pak 5. vývodem bude vstup tlačítka. Není tedy třeba používat mikrokontroler s 16 I/O piny (pouzdro s 18, nebo 20 vývody), ale bude stačit s 6 I/O piny (tedy pouzdro s 8 vývody).