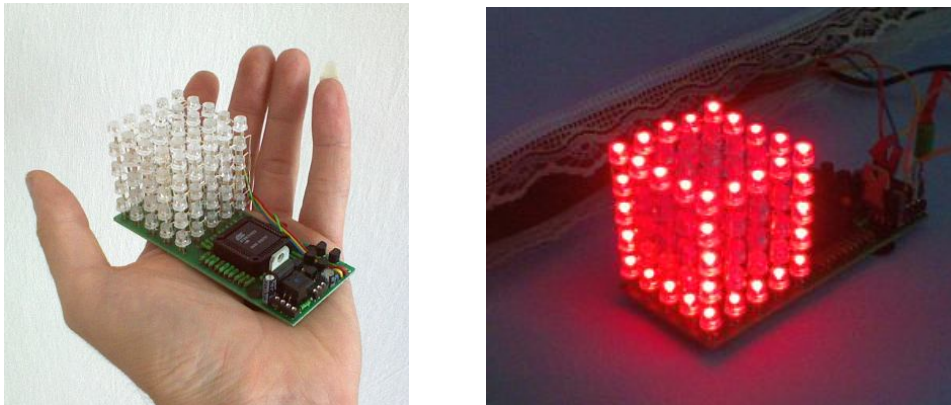


LED CUBE

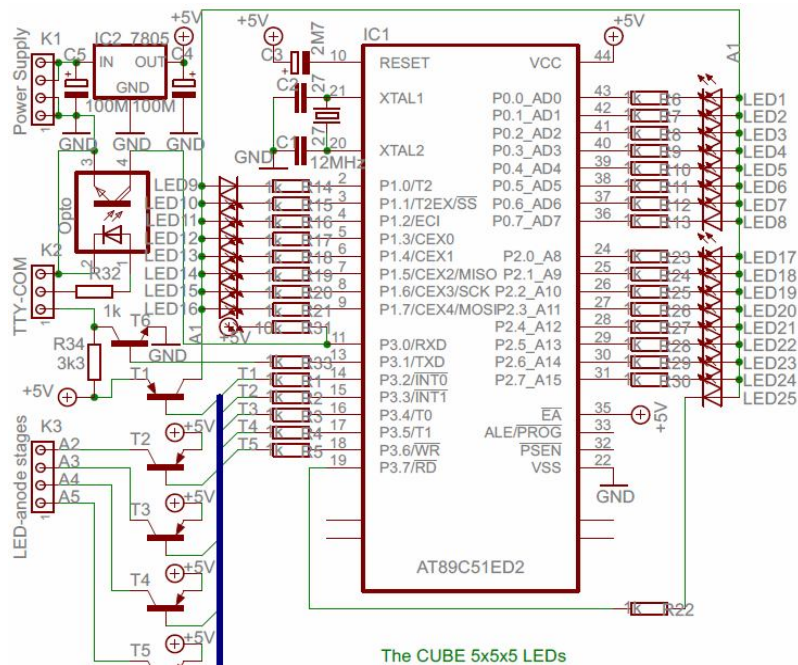
(schéma + software: Borik Zlatović – 2010, konštrukcia: Hannibal / Kowalczyk – 2012)

Pod týmto názvom sa neskrýva nič iné, než jednoduchá aplikácia „kocky“, ktorú určite už každý z fanov mikroelektroniky videl na YOUTUBE v rôznych (i RGB) modifikáciách. Mňa osobne táto vec značne zaujala, pretože som nikde nenašiel konkrétnu a **hlavne jednoduchú schému**, ako toto postaviť. Konštrukcia B. Zlatovića ale poskytuje **ucelený návod užívateľom programovateľnej aplikácie kocky**, preto som vďačný poliakovi Olegovi Kowalczykovi, ktorý mi to preposlal mailom...

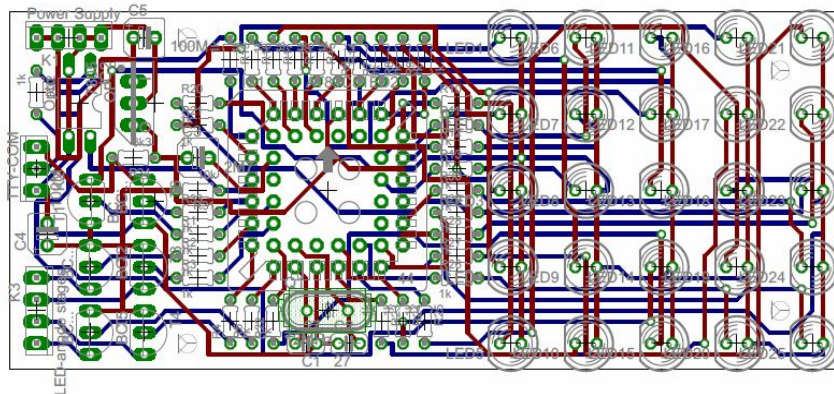


obr.1 – miniatúrna konštrukcia kocky 5x5x5 LED

Schéma je jednoduchá, všetky súčiastky sú rozmiestnené na jednom plošnom spoji, pretože pôvodný zámer Borika Zlatovića bolo umiestniť a používať kocku v osobnom aute. Ako prvé si osadíte sokel pre MCU, potom stabilizátor, kondenzátory, všetky odpory a nakoniec, konektory pre napájanie a TTY. Optiku som použil hocijakú starú TESLA, čo som vybral zo starých plošákov, tranzistory to samé – predhistorické TESLA typy: KC... (pnp - 307, 8, 9 a npn - 508). Potom dosku oživate, pripojíte na sériový port, zasuniete do päťice MCU a vyskúšajte si funkčnosť komunikácie s programom „FLIP“, (download na: www.atmel.com). Napáľte program „LED_CUBE_v0.2b.hex“ do MCU a zrušte switch „BLJB“ (pripadne X2). MCU vyberte von: do päťice sa vloží až po zaletovaní LED diód.

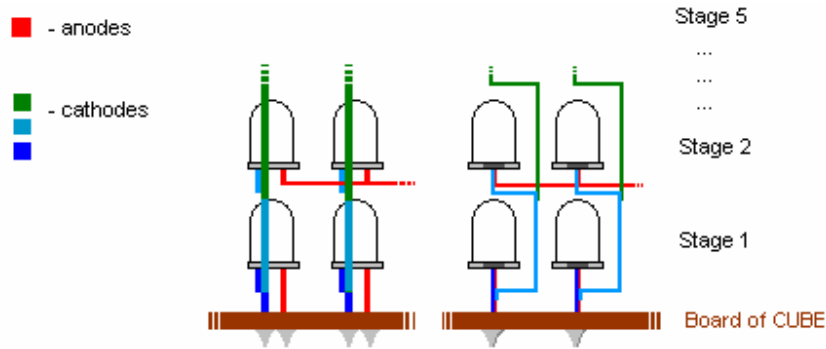


obr.2 – schéma zapojenia kocky



obr.3 – doska konštrukcie kocky (Oleg Kowalczuk, EAGLE)

Doska je navrhnutá pre LEDky priemeru 3mm. Ja som použil, žiaľ, 5mm diody a z každej som urezal asi polovicu dĺžky. Malo to 2 dôvody: prvý bol ten, že som ich zohnal po 5 centov / kus a druhý, že sklené zapuzdrenie LED spôsobuje smerovanie svetla rovno, čo dosť blbo ožiarovalo diody v radoch nad sebou a ich viditeľnosť z boku bola slabá. Ak ich ale orežete, brúsená plocha potom svetlo rozptyľuje a viditeľnosť je výborná... LEDky použite vysokosvietiace (cca 5000 – 8000cd), nie obyčajné: slabo by svietili skrz prúdové limity portov, obmedzené odpormi s hodnotami až 1kΩ. Displej sa skladá z 5 „etáží“ po 25 LED. Prvá etáž je navrhnutá už priamo na doske, no ďalšie je nutné „vymodelovať“ z diod ručne a pripájať na nožičky diod prvej etáže – na katódy. Každú nasledujúcu etáž potom pripájate na katódy LED predošlej. 25 anód každej etáže je spojených dokopy a sú pripojené dráťkami na výstupy A2 (etáž 2), A3 (etáž 3)... až A5, ktorá je najvyššie.



obr.4 – možné „viazanie“ LED jednotlivých etáží podľa Borika Zlatoviča

Program kocky je **BETA verzia**, čo znamená, že **nie je možné urobiť vlastné sekvencie**, ale keďže „kocka“ obsahuje niekoľko už vytvorených, tak to nie je vyslovene potrebné. Vyššia verzia mohla prepnúť kocku do „edit-mode“. Ten mal vyhradenú časť FLASH pre užívateľa, do ktorej sa pomocou terminálového programu vložili jednotlivé „frames“ (sorry na názov, ale neviem to nazvať), ktoré sa editovali klávesnicou PC: stláčaním tlačidiel sa rozsvetcovali negovaním bitov v MCU LEDky jednotlivých etáží a užívateľ sledoval okamžité reakcie priamo na kocke. Následne sa potvrdením uložila celá „frame“ a po nej nasledovali ďalšie. Po zadefinovaní konca sekvencie (poslednej frame), sa ešte omarkoval začiatok automatickým zapamätaním adresy. Všetko prebiehalo priamo v procesore, ktorý zrejme využíval interné IAP rutiny a nebol k tomu okrem terminálového programu potrebný žiadny iný software...

```

=====
the CUBE - sequency editor
=====

A      adr = define beqin of user sequency
M      auto-mark range user sequency (if sequency finished)
E      insert end of sequency (apply to last used frame..)
R      rate value (timing frames)
P      play sequency

1 2 3 4 5      0 0 0 0 0 - editing LED (actual stage)
q w e r t      0 0 0 0 0
a s d f g      0 0 0 0 0
z x c v b      0 0 0 0 0
n m , . /      0 0 0 0 0

[          stage decrease
          stage increase
]

enter      save 1 frame to FLASH
space      load 1 frame from FLASH and adr increment
backspace  load 1 frame from FLASH and adr decrement
Q          leave editor
    
```

obr.5 – ukážka časti z editoru „užívateľských sekvencií“ vo vyššej verzii programu tejto aplikácie...

Verzia 0.2b má aspoň možnosť vytvoriť postupnosť celých, autorom navrhnutých sekvencií, čo plne postačuje už tým pádom, že kocka obsahuje sekvencie písmen celej abecedy. Tak je možné spraviť program s ľubovoľným poradím a počtom opakovania sa vybraných sekvencií a vložení vlastného textu. Celé programovanie prebieha zadávaním kódov na klávesnici PC použitím TTY.

CODE	SEQUENCY	CODE	SEQUENCY	CODE	SEQUENCY		
00	X/Y/Z move frames 1+3+5	10	peripheral frame move	20	X/Y/Z U-block rotate		
01	X/Y plate rotate	11	X/Y/Z frame blinking	21	Z mill		
02	outline cube up-down	12	X rotation 4x4 squares	22	Y mill		
03	X 8-edge rotate	13	Z rotation 4x4 squares	23	X/Y frame-worm		
04	X/Y/Z 3x3x3cube move	14	Y rotation 4x4 squares	24	A symbol		
05	X/Y/Z 3x3x3cube resize	15	matrix effect	25	B symbol		
06	X/Y/Z frame move/rotate	16	outline cube down-up	26	C symbol		
07	X 8-points anti-rotate	17	X/Y/Z move frames	27	D symbol		
08	full cube blinking	18	X fountain	28	E symbol		
09	X twister	19	Y fountain	29	F symbol		
0a	Z/Y waves	1a	tetris	2a	G symbol		
0b	X/Y/Z edge-lines	1b	X/Y diagonals	2b	H symbol		
0c	Y square scope	1c	X/Y/Z full cube resize	2c	I symbol		
0d	Z triangle scope	1d	Z fountain	2d	J symbol		
0e	X/Y/Z square tunnel	1e	Z 8-points anti-rotate	2e	K symbol		
0f	dual X/Y/Z edge-lines	1f	Y 8-points anti-rotate	2f	L symbol		
30	M symbol	40	2 symbol	50	: symbol	60	& symbol
31	N symbol	41	3 symbol	51	" symbol	61	^ symbol
32	O symbol	42	4 symbol	52	? symbol	62	/ symbol
33	P symbol	43	5 symbol	53	! symbol	63	~ symbol
34	Q symbol	44	6 symbol	54	@ symbol		
35	R symbol	45	7 symbol	55	< symbol	64	sequences of this
36	S symbol	46	8 symbol	56	> symbol	..	codes is not available
37	T symbol	47	9 symbol	57	(symbol	..	in v0.2beta software
38	U symbol	48	- symbol	58) symbol	ff	
39	V symbol	49	+ symbol	59	[symbol		
3a	W symbol	4a	- symbol	5a] symbol		
3b	X symbol	4b	* symbol	5b	_ symbol		
3c	Y symbol	4c	+ symbol	5c	symbol		
3d	Z symbol	4d	= symbol	5d	# symbol		
3e	0 symbol	4e	, symbol	5e	\$ symbol		
3f	1 symbol	4f	. symbol	5f	% symbol		

obr.6 – „hexa“ kódy jednotlivých sekvencií kocky

Programovanie je jednoduché: nastavte si terminálový program na sériový port (napr. COM1, 9600/N/8/1) a pripojte kocku káblom pomocou vodičov RXD, TXD a GND (piny 2, 3 a 5 konektoru portu). Po zapnutí napájania je kocka v stave vykonávania programu, ktorý načítava z vnútornej pamäte procesora. Klepnite do klávesnice a na termináli sa objaví nápis:

```
*The_CUBE v0.2b*
?
```

Kocku ovládate nasledovnými klávesami: „enter“ – editovanie jedného kroku programu, „i“ – inkrement adresy, „d“ – dekrement adresy, „l“ – listovanie programu, „c“ – clear, zmazanie celého programu a „g“ – go, opätovné spustenie programu.

napr.

```
*The_CUBE v0.2b*
?          enter (po stlačení môžete vložiť dáta na aktuálne nastavenej adrese)

*The_CUBE v0.2b*
00: seq=FF?          enter (otáznik naznačuje, či chcete pokračovať, ak nie, stlačte inú klávesu...)

*The_CUBE v0.2b*
00: seq=FF<--          1, 6 (klávesami 0-9 / a-f vložíte nejaký kód podľa tabuľky sekvencií kocky)

*The_CUBE v0.2b*
00: seq=FF<--16 run=-          1 (klávesami 0-9 vložíte číslo, koľkokrát sa má sekvencia zopakovať)

*The_CUBE v0.2b*
00: seq=FF<--16 run=-1 spd=-          7 (klávesami 0-9 vložíte číslo, akou rýchlosťou sa má sekvencia vykresľovať)

*The_CUBE v0.2b*
00: seq=FF<--16 run=-1 spd=-7
01: seq=FF?          enterom môžete pokračovať ďalej v editovaní programu...

*The_CUBE v0.2b*
00: seq=16 run=1 spd=7          klávesou list, sa vypíše celý program (výpis možno prerušiť ľubovoľnou klávesou)
01: seq=02 run=1 spd=7
02: seq=00 run=2 spd=9
03: seq=FF run=- spd=-
04: seq=FF run=- spd=-
```

Keď máte program hotový, spustíte ho klávesou g (go). Editor kocky je veľmi jednoduchý, preto neobsahuje možnosť posunúť a vložiť, či naopak – odmazat' krok programu niekde v strede pamäte. Preto ak máte napísaný väčší úsek programu a nechcete, aby sa niektorá z predtým vložených sekvencií vykonala, zmažte ju vloženíím kódu z rozsahu „64-ff“: kódy sekvencií nad 64 sú automaticky preskočené. To samé platí i pri vložení čísla 0 do pozícií „run=“, či „spd=“. Ak chcete zmazať celý program, stlačte klávesu c (clear) a počkajte – mazanie pamäte trvá procesoru asi 8 sekúnd... Na záver – výpis programu, z ktorého je urobené video:

00: seq=16 run=1 spd=7	3F: seq=35 run=1 spd=7
01: seq=02 run=1 spd=7	40: seq=36 run=1 spd=7
02: seq=00 run=2 spd=7	41: seq=4F run=1 spd=7
03: seq=0A run=1 spd=7	42: seq=26 run=1 spd=7
04: seq=2F run=1 spd=7	43: seq=32 run=1 spd=7
05: seq=28 run=1 spd=7	44: seq=30 run=1 spd=7
06: seq=27 run=1 spd=7	45: seq=FF run=- spd=-
07: seq=48 run=1 spd=7	46: seq=FF run=- spd=-
08: seq=26 run=1 spd=7	47: seq=FF run=- spd=-
09: seq=38 run=1 spd=7	48: seq=FF run=- spd=-
0A: seq=25 run=1 spd=7	49: seq=FF run=- spd=-
0B: seq=28 run=1 spd=7	4A: seq=07 run=9 spd=9
0C: seq=FF run=- spd=-	4B: seq=1E run=5 spd=9
0D: seq=FF run=- spd=-	4C: seq=1E run=2 spd=7
0E: seq=FF run=- spd=-	4D: seq=1E run=1 spd=5
0F: seq=FF run=- spd=-	4E: seq=1E run=1 spd=4
10: seq=10 run=1 spd=7	4F: seq=1E run=1 spd=3
11: seq=16 run=1 spd=7	50: seq=1E run=1 spd=2
12: seq=02 run=1 spd=7	51: seq=1E run=1 spd=1
13: seq=21 run=9 spd=7	52: seq=1F run=1 spd=1
14: seq=22 run=9 spd=7	53: seq=1F run=1 spd=2
15: seq=09 run=2 spd=9	54: seq=1F run=1 spd=3
16: seq=09 run=1 spd=8	55: seq=1F run=1 spd=4
17: seq=09 run=1 spd=7	56: seq=1F run=1 spd=5
18: seq=09 run=1 spd=6	57: seq=1F run=1 spd=6
19: seq=09 run=1 spd=5	58: seq=1F run=1 spd=7
1A: seq=09 run=1 spd=4	59: seq=1F run=1 spd=8
1B: seq=09 run=1 spd=3	5A: seq=1F run=6 spd=9
1C: seq=09 run=1 spd=2	5B: seq=15 run=1 spd=7
1D: seq=09 run=1 spd=2	5C: seq=18 run=2 spd=6
1E: seq=09 run=1 spd=4	5D: seq=FF run=- spd=-
1F: seq=09 run=1 spd=6	5E: seq=FF run=- spd=-
20: seq=09 run=1 spd=8	5F: seq=FF run=- spd=-
21: seq=09 run=1 spd=9	60: seq=0C run=2 spd=9
22: seq=01 run=1 spd=9	61: seq=0D run=2 spd=9
23: seq=01 run=1 spd=7	62: seq=16 run=1 spd=7
24: seq=01 run=1 spd=5	63: seq=17 run=3 spd=9
25: seq=01 run=1 spd=3	64: seq=1C run=2 spd=9
26: seq=01 run=3 spd=1	65: seq=02 run=1 spd=7
27: seq=08 run=9 spd=5	66: seq=06 run=2 spd=7
28: seq=13 run=3 spd=1	67: seq=23 run=3 spd=9
29: seq=13 run=2 spd=2	68: seq=FF run=- spd=-
2A: seq=13 run=1 spd=3	69: seq=FF run=- spd=-
2B: seq=13 run=1 spd=4	6A: seq=FF run=- spd=-
2C: seq=13 run=1 spd=5	6B: seq=FF run=- spd=-
2D: seq=13 run=1 spd=6	6C: seq=FF run=- spd=-
2E: seq=13 run=2 spd=7	6D: seq=FF run=- spd=-
2F: seq=13 run=2 spd=9	6E: seq=FF run=- spd=-
30: seq=14 run=6 spd=9	6F: seq=FF run=- spd=-
31: seq=0B run=1 spd=7	70: seq=24 run=1 spd=7
32: seq=16 run=1 spd=7	71: seq=37 run=1 spd=7
33: seq=02 run=1 spd=7	72: seq=46 run=1 spd=7
34: seq=3A run=3 spd=7	73: seq=47 run=1 spd=7
35: seq=4F run=1 spd=7	74: seq=26 run=1 spd=7
36: seq=30 run=1 spd=7	75: seq=43 run=1 spd=7
37: seq=26 run=1 spd=7	76: seq=3F run=1 spd=7
38: seq=32 run=1 spd=7	77: seq=28 run=1 spd=7
39: seq=31 run=1 spd=7	78: seq=27 run=1 spd=7
3A: seq=37 run=1 spd=7	79: seq=40 run=1 spd=7
3B: seq=35 run=1 spd=7	7A: seq=1B run=1 spd=9
3C: seq=32 run=1 spd=7	7B: seq=1A run=2 spd=9
3D: seq=2F run=2 spd=7	7C: seq=04 run=3 spd=9
3E: seq=28 run=1 spd=7	7D: seq=0F run=1 spd=9