Technická univerzita v Liberci/Fakulta umění a architektury
Katedra výtvarných umění/Vizuální komunikace - Digitální média
Bakalářská práce: Sem tam

David Šmíd/Liberec 2010
Vedoucí práce: Doc. Stanislav Zippe

//prohlášení

"Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně."

David Šmíd

//poděkování

Děkuji panu Doc. Stanislavu Zippemu a Mgr. Jaroslavu Prokešovi za hodnotné rady a odborné vedení během mé práce.

//obsah

Vizuálně aktivní dílo - to je pojem, který prostupuje celou touto diplomovou prací. Jde o barevně pulzující světelné vertikály - na bílé projekční ploše se prolínají, kombinují a variují barevné "skvrny". Jsou zde neustálé vizuální proměny v rozmístění barevnosti i množství těchto barevných "skvrn", které může i člověk/divák svojí přítomností ovlivňovat. Právě vlastní aktivita (vlastní život objektu) v kombinaci s možností objekt ovlivnit je dalším silným motivem mé práce.

V této souvislosti bych zde chtěl zmínit jméno Radek Kratina, jehož díla - variabily, na mne zapůsobily. Jsou to objekty (systémy), které mají mnoho poloh, uspořádání a uzpůsobení, mají mnoho variací a hlavě jsou hravé - divák má možnost jejich "tvary" libovolně měnit a tak vytvářet stále něco nového.

Jeho díla - varibily, se začínají rozvíjet od šedesátých let. Ve své podstatě mají blízko k jeho předchozí tvorbě dětských hraček, v nichž si vyzkoušel princip proměny v prostorovém řešení (Figurky a kostky měly děti přimět k hravosti a také rozvíjet jejich smysl pro uspořádání elementů podle určitých pravidel).
Této proměnlivosti je využito právě ve variabilech. Prvním z těchto proměnlivých objektů je dřevěný rámeček vyplněný vypálenými zápalkami, které tvoří reliéf. Reliéf proměnlivý, protože jeho struktura se mění tlakem prstů na jeho povrch. Později následoval objekt, v němž lze horizontálně posunovat tyčinkami, rozdělenými na bílá a červená pole. *Potom přicházely další nápady. Sochař barvil tyčinky jen na jedné polovině, aby k proměně došlo také jejich otáčením. Pak tyčky instaloval do několika polí nebo na více os. Některé objekty se rozvíjely do volného prostoru. Tím se utvořily základní řady umělcovy tvorby. Koncem šedesátých let se rozhodl, že přestane užívat dřevo. Nesplňovalo totiž některé jeho požadavky. Nebylo možné ho dostatečně přesně opracovat a také se trvale sesychalo. Proto přešel ke kovům. Jejich obrábění však vyžadovalo dokonalé technické vybavení. Již nemohl realizovat své představy sám v panelákovém pokoji - ateliéru. Rozkresloval projekty a podle nich nechával své variabily zhotovit. Nejdříve užíval železo, později také mosaz a lehký dural. Povrch kovů různě upravoval - leštil, černil, nikloval, chromoval, matoval, členil vruby. Tyto úpravy nahradily dřívejší barevné odlišení na dřevěných variabilech.* (Radek Kratina 1928-1999, Idea con variazioni, Jan Sekera, Jiří Machalický, Jan Kříž, Radek Kratina, Josef Hlaváček, k výstavě v ČMVU v Praze od 5. 4. do 24. 5. 2000)

Druhým jménem a zároveň jistým protikladem, zvláště pak v oboru působení v běžném životě, je Frank Joseph Malina - vědec, raketový konstruktér. Je na tom zajímavý právě vstup člověka - vědce do pole umění. Frank Malina se narodil v Texasu, přesto však i krátký čas strávil v Československé republice. Kde s jeho rodiči (lidoví muzikanti z Moravy) několik let žil. Po návratu z Moravy dokončil střední školu a poté se přihlásil ke studiu leteckého inženýrství na Caltech.... Později získal doktorský titul u profesora Theodora von Kármána, slavného aerodynamika maďarského původu, s nímž spolupracoval a stal se úspěšným konstruktérem první americké výškové rakety WAC Corporal. V roce 1947 ale přerušil svoji vědeckou kariéru a odjel do Evropy, kde začal pracovat v nově založené organizaci UNESCO (Zde měl na starosti program rozvoje vědy a výzkum suchých oblastí naší planety). Svoji uměleckou kariéru započal v padesátých letech. V roce 1968 založil Malina revue pro umění, vědu a technologie a pojmenoval ji příznačně Leonardo. Vznikla tak zcela unikátní tradice psaní o uměleckých experimentech na hranicích s vědou a technologiemi. Umělci i vědci se na stránkách tohoto časopisu, vydávaného MIT Press, vzájemně informují o prolínání umělecké a vědecké kreativity v kontextu nových technologií. Umělci popisují své experimenty po vzoru vědeckých článků, jejichž smyslem je především nacházet nové cesty vývoje umění, nikoli svět umění uzavírat a uchylovat se k hermetickým, tajuplným „interpretacím".

Frank Malina patří mezi zakladatelské osobnosti a průkopníky kinetického umění. Jeho fascinující životní dráha v sobě ukrývá příběh experimentálního umělce, který vynalezl originální světelně kinetické systémy, svého druhu pohyblivé abstraktní malby, stejně jako příběh světoznámého vědce, označovaného za "kmotra" amerického vesmírného programu.

Jas diody - jas lze ovlivnit pomocí rychlého zapínání a vypínání diody - tzv. PWM (Pulse Width Modulation), jas je tedy určen poměrem času zapnuté a vypnuté LED diody. Periodu (čas, kdy je dioda vypnutá + čas, kdy je zapnutá) musíme zvolit dostatečně krátkou, aby nedocházelo k přílišnému blikání diody - že lidské oko pozná, že LED dioda nezhasíná plynule - úbytek napětí, ale právě poměrem jednotlivých dob svícení/zhasnutí.Experimentálně jsme určili jako nejlepší rozdělení intenzity na 20 jasových úrovní (v závislosti na možnostech zvolených integrovaných obvodů).

Výsledná barva LED diody - je použita LED dioda RGB - jedna LED dioda obsahuje tři emitovatelné barvy - červenou, zelenou a modrou. (Každá barva se ovládá příslušnou "nožičkou" (katodou)) Aby bylo dosaženo širokého spektra barev, jednotlivé barvy (kanály RGB) se míchají. I zde je využito "PWM" modulace.

Spínání pomocí IR LED čidla - spínání je prováděno pomocí IR LED diody a fotocitlivých IR diod(fotocitlivý IR tranzistor). Princip je založen na odrazu paprsků z vysílací IR diody do IR přijímače od předmětu umístěného před "detektorem". Pro snížení chybovosti čidla vůči okolnímu prostředí (denní světlo, žárovky a zářivky IR záření vysílají -> jsou tedy rušivými elementy) je použito následující "opatření" - frekvenční modulace, vysílaný určitý kód. To znamená, že program se spustí pouze tehdy, obdrží-li přijímací dioda určitý kód. Specifický kód v tomto případě je 101 (je zvolen pouze jednoduchý kód. Složitější (přesnější) by měl nepříznivý vliv na celkovou rychlost operací, které ovládací čip provádí - tedy zvýšení prodlev ve svitu/zhasnutí LED diod a následné rozblikání) ...nejprve je tedy změřené napětí fotocitlivé diody při zapnuté diodě (U_1), poté při vypnuté (U_2) a naposled znovu při zapnuté (U_3). Pokud se rozdíl napětí rozsvícené a zhasnuté diody dostatečně změní (stanovené prahem) a zároveň se téměř nezmění napětí změřené při prvním a posledním měření (pokaždé zapnutá emitující dioda), pak lze s vysokou pravděpodobností určit, že před detektorem se nachází nějaký objekt -> impulz ke spuštění programu.

Řízení celého "systému" - zprostředkovává jednoduchý obvod, složený z ovládacího čipu (v tomto případě Atmel ATMega328 s nahraným bootloaderem (Arduino) - pro jednodušší programování), několika doprovodných součástek pro čip (16MHz krystal, kondenzátory a odpory) ...a několika posuvných registrů, které umožňují ovládání (poskytují dostatek výstupů pro velký počet ovládaných LED diod).

Čipy jsou programovány v intuitivním prostředí založeném na jazyku WIRING (podobný jazyku C#)

```
#define COLORS 3                        // pocet barevnych kanalu diody
#define DIODES 16                       // pocet pouzitych RGB diod
#define MAX_LEVEL 20                    // pocet urovni osvetleni
#define DROP_LENGTH 20                  // delka jedne kapky

byte clockPin = 2;                      // prirazeni hodinoveho signalu pro posuvne registry na pin
byte checkPin = 13;                     // prirazeni pinu signalni diody
byte dataPin[] = {6, 7, 8};             // prirazeni datovych signalu pro jednotlive barevne kanaly
byte latchPin[] = {11, 10, 9};          // prirazeni posuvnych signalu pro jednotlive barevne kanaly
byte irIn = 12;                         // prirazeni pinu z externiho IR cidla
byte irOut = 4;                         // prirazeni pinu pro IR diodu pro interni IR cidla
byte PHOTOS[2] = {0, 1};                // prirazeni pinu pro interni IR cidla

boolean irTop = false, irBottom = false;        // promene urcujici zdali jsou sepnuty jednotlive IR cidla

// pole urovni osvetleni pro kazdy barevny kanal kazde diody, co je v poli, to se v full_cykl vykresli
byte data[16][3] = {{0,0,0},{0,0,0},{0,0,0},{0,0,0},
                    {0,0,0},{0,0,0},{0,0,0},{0,0,0},
                    {0,0,0},{0,0,0},{0,0,0},{0,0,0},
                    {0,0,0},{0,0,0},{0,0,0},{0,0,0}};

// rozdeleni kapky do jednotlivych fazi
float dropb[DROP_LENGTH][5] = {{0,0,19,0,0},
                    {0,1,18,1,0},
                    {0,2,17,2,0},
                    {0,3,17,3,0},
                    {1,4,16,4,1},
                    {1,5,16,5,1},
                    {1,6,14,6,1},
                    {1,7,14,7,1},
                    {1,6,12,6,1},
                    {2,6,10,6,2},
                    {2,5,9,5,2},
                    {2,5,8,5,2},
                    {2,5,7,5,2},
                    {1,5,6,5,1},
                    {1,4,5,4,1},
                    {1,3,3,3,1},
                    {1,1,2,1,1},
                    {0,1,1,1,0},
                    {0,0,1,0,0},
                    {0,0,0,0,0}
                   };
/** funcke setup
 *
 *  inicializace pinu atmelu
 */
void setup() {
  pinMode(clockPin, OUTPUT);
  pinMode(checkPin, OUTPUT);
  pinMode(irOut, OUTPUT);
  pinMode(irIn, INPUT);
  for(int i= 0; i < COLORS; i++) {
    pinMode(latchPin[i], OUTPUT);
    pinMode(dataPin[i], OUTPUT);
  }
  full_cykl();
  randomSeed(analogRead(0));
}

/** funkce loop
 *
 *  funkce opakujici se od spusteni do konce .. ve skutecnosti prijde jen jednou, pak porad bezi nobody
 */

void loop() {
  nobody();                     // program pro okamzim kdy neni nikdo pritomen
}

/** funkce boolean analogIR()
 *
 *  rozhoduje zdali byl zaznamenan pohyb na nekterem IR cidlu sloupu
 *
 *  parametry SAMPLE .. urcuje pocet vzorku hodnot IR cidla, ktere pak jsou zprumerovany
 *           K     .. prah, ktery musi byt prekrocen mezi hodnotou na cidle pri zapnute a vypnute IR diode
 *           D     .. prah, ktery nesmi byt prekrocen mezi dvemi merenimi pri zapnute diode
 */
```

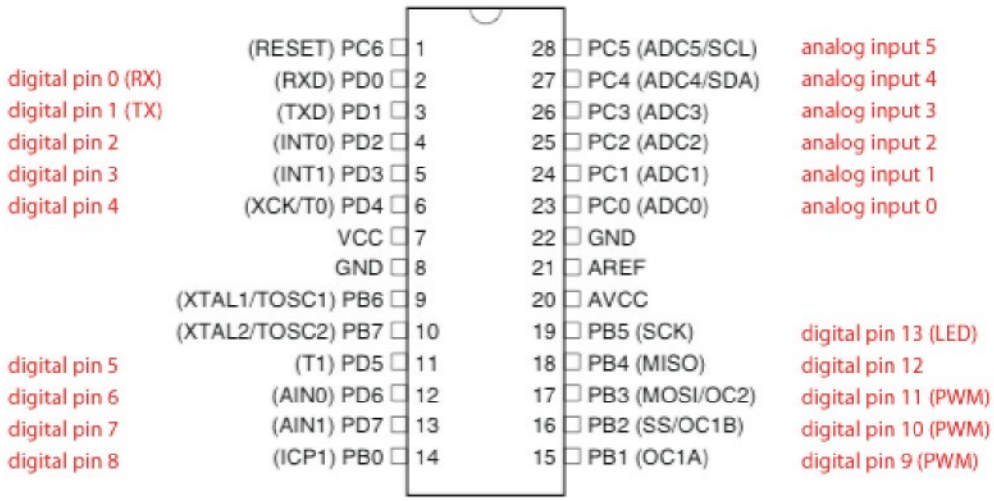
```
#define SAMPLE 2
#define K 50
#define D 2

boolean analogIR() {
  int dts[2][SAMPLE];

  // nacitani 1. vzorku pri rozsvicene IR diode
  digitalWrite(irOut, HIGH);
    for(int i = 0; i<SAMPLE; i++) {
      dts[0][i] = analogRead(PHOTOS[0]);
      dts[1][i] = analogRead(PHOTOS[1]);
    }
    float val1[] = {0,0};
    for(int i = 0; i<SAMPLE; i++) {
      val1[0] += dts[0][i];
      val1[1] += dts[1][i];
    }
    val1[0] /= SAMPLE;
    val1[1] /= SAMPLE;

  // nacitani 2. vzorku pri zhasnute IR diode
  digitalWrite(irOut, LOW);
    for(int i = 0; i<SAMPLE; i++) {
      dts[0][i] = analogRead(PHOTOS[0]);
      dts[1][i] = analogRead(PHOTOS[1]);
    }
    float val2[] = {0,0};
    for(int i = 0; i<SAMPLE; i++) {
      val2[0] += dts[0][i];
      val2[1] += dts[1][i];
    }
    val2[0] /= SAMPLE;
    val2[1] /= SAMPLE;

  // nacitani 3. vzorku pri rozsvicene IR diode
  digitalWrite(irOut, HIGH);
    for(int i = 0; i<SAMPLE; i++) {
      dts[0][i] = analogRead(PHOTOS[0]);
      dts[1][i] = analogRead(PHOTOS[1]);
    }
    float val3[] = {0,0};
    for(int i = 0; i<SAMPLE; i++) {
      val3[0] += dts[0][i];
      val3[1] += dts[1][i];
    }
    val3[0] /= SAMPLE;
    val3[1] /= SAMPLE;

  digitalWrite(irOut, LOW);

  // IR cidlo "sepnuto", pokud mezi rozsvicenou a zhasnutou diodou je prekrocen prah K, ale mezi vzorky
  //   rozsvicenych diod neni prekrocen prah D
  irTop = ((val2[0]-val1[0])>K) && ((val1[0] - val3[0]) < D) && ((val3[0] - val1[0]) < D);
  irBottom = ((val2[1]-val1[1])>K) && ((val1[1] - val3[1]) < D) && ((val3[1] - val1[1]) < D);

  return irTop || irBottom;
}

/** funkce nobody
 *
 *  sem tam jedna bila kapka (okamzik kdy nikdo neni v oblasti)
 */

void nobody() {
    byte level = random(MAX_LEVEL/2);
    byte phase = 0;
    byte pos = random(DIODES-3);
    while(true) {
      if(phase>=DROP_LENGTH) {
        if(random(50)>0)
          phase = DROP_LENGTH;
        else {
          phase = 0;
          level = random(MAX_LEVEL/2);
          pos = random(DIODES-3);
        }
```

```
} else {
      for(byte i = 0; i < 5; i++) {
        byte val = round(dropb[phase][i]/MAX_LEVEL*level);
        data[i+pos][0] = data[i+pos][1] = data[i+pos][2] = val; // = data[i+pos][1] = data[i+pos][2]
      }
      phase++;
    }
    for(byte w = 0; w<3; w++) {
      full_cykl();
      if(irTop || irBottom || analogIR() || digitalRead(irIn)==HIGH)
        anybody();
      analogIR();
    }
  }
}

/** funkce anybody
 *
 *  vice bilych kapek (okamzik kdyz se nekdo pasivni nachazi v oblasti)
 *
 *  parametr ANYBODY_ITER .. urcuje pocet iteraci, po kterych se ma program anybody ukoncit
 */

#define ANYBODY_ITER (((random(10))*20)+400)
void anybody() {
  byte level[] = {random(MAX_LEVEL),random(MAX_LEVEL)};
  byte phase[] = {DROP_LENGTH, DROP_LENGTH};
  byte pos[] = {random(DIODES-3),random(DIODES-3)};
  int Zrand = random(5);

  unsigned int iter = ANYBODY_ITER;
  while(iter>0) {
    for(byte i = 0; i < DIODES; i++)
      data[i][0] = data[i][1] = data[i][2] = 0;
    for(byte r = 0; r < 2; r++) {
      if(phase[r]>=DROP_LENGTH) {
        if(random(4)>0)
          phase[r]= DROP_LENGTH;
        else {
          phase[r]= 0;
          level[r] = random(MAX_LEVEL);
          pos[r] = random(DIODES-3);
        }
      } else {
        for(byte i = 0; i < 5; i++) {
          byte val = round(dropb[phase[r]][i]/MAX_LEVEL*level[r]);
          data[i+pos[r]][0] += val;
          data[i+pos[r]][1] += val;
          data[i+pos[r]][2] += val; // = data[i+pos][1] = data[i+pos][2]
        }
        phase[r]++;
      }
    }
    for(byte w = 0; w<3; w++) {
      full_cykl();
      if(digitalRead(irIn)==HIGH)
        iter = ANYBODY_ITER;
      if(irTop || irBottom || analogIR()) {
        sensor();
        return;
      }
      if(Zrand < 1) {
        sensorF();
      }
      analogIR();
    }
    iter--;
  }
}

/** funkce sensor
 *
 *  program vice barevnych kapek (okamzik sepnuti detektoru)
 *
 *  parametr SENSOR_ITER .. urcuje pocet iteraci po kterych se ma program sensor ukoncit
 */
```

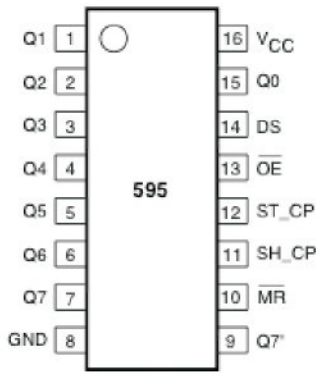
```
#define SENSOR_ITER 500
void sensor() {
  byte level[3][3] = {{0,0,0},{0,0,0},{0,0,0}};
  byte phase[] = {DROP_LENGTH, DROP_LENGTH};
  byte pos[] = {random(DIODES-3),random(DIODES-3)};
  unsigned int iter = SENSOR_ITER;
  while(iter>0) {
    if(iter>300)
      digitalWrite(checkPin, HIGH);
    else
      digitalWrite(checkPin, LOW);
    for(byte i = 0; i < DIODES; i++)
      data[i][0] = data[i][1] = data[i][2] = 0;
    for(byte r = 0; r < 2; r++) {
      if(phase[r]>=DROP_LENGTH) {
        if((iter>300 && random(4)==0) || (iter>150 && random(16)==0) || (iter>50 && random(64)==0)) {
          phase[r]= 0;
          level[r][0] = random(MAX_LEVEL);
          level[r][1] = random(MAX_LEVEL);
          level[r][2] = random(MAX_LEVEL);
          pos[r] = random(DIODES-3);
        } else
          phase[r]= DROP_LENGTH;
      } else {
        for(byte i = 0; i < 5; i++) {
          data[i+pos[r]][0] += round(dropb[phase[r]][i]/MAX_LEVEL*level[r][0]);
          data[i+pos[r]][1] += round(dropb[phase[r]][i]/MAX_LEVEL*level[r][1]);
          data[i+pos[r]][2] += round(dropb[phase[r]][i]/MAX_LEVEL*level[r][2]);
        }
        phase[r]++;
      }
    }
    for(byte w = 0; w<5; w++) {
      full_cykl();
      if(irTop || irBottom || analogIR())
        iter= SENSOR_ITER;
    }
    analogIR();
    iter--;
  }
}

//Forte funkce

void sensorF() {
  byte level[3][3] = {{0,0,0},{0,0,0},{0,0,0}};
  byte phase[] = {DROP_LENGTH, DROP_LENGTH};
  byte pos[] = {random(DIODES-3),random(DIODES-3)};
  unsigned int iter = SENSOR_ITER;
  while(iter>0) {
    if(iter>300)
      digitalWrite(checkPin, HIGH);
    else
      digitalWrite(checkPin, LOW);
    for(byte i = 0; i < DIODES; i++)
      data[i][0] = data[i][1] = data[i][2] = 0;
    for(byte r = 0; r < 3; r++) {
      if(phase[r]>=DROP_LENGTH) {
        if((iter>300 && random(4)==0) || (iter>150 && random(16)==0) || (iter>50 && random(64)==0)) {
          phase[r]= 0;
          level[r][0] = random(MAX_LEVEL);
          level[r][1] = random(MAX_LEVEL);
          level[r][2] = random(MAX_LEVEL);
          pos[r] = random(DIODES-3);
        } else
          phase[r]= DROP_LENGTH;
      } else {
        for(byte i = 0; i < 5; i++) {
          data[i+pos[r]][0] += round(dropb[phase[r]][i]/MAX_LEVEL*level[r][0]);
          data[i+pos[r]][1] += round(dropb[phase[r]][i]/MAX_LEVEL*level[r][1]);
          data[i+pos[r]][2] += round(dropb[phase[r]][i]/MAX_LEVEL*level[r][2]);
        }
        phase[r]++;
      }
    }
    for(byte w = 0; w<5; w++) {
      full_cykl();
```

```
 if(irTop || irBottom || analogIR())
        iter= SENSOR_ITER;
     }
     analogIR();
     iter--;
   }
}


/** funkce full_cykl
 *
 *  vykresli jeden cely cykl (vsechny faze ... opakuje se podle poctu levelu)
 *
 */

void full_cykl()  {
  for(int i = 0; i < MAX_LEVEL; i++)
    cykl(i);
}

/** funkce cykl(int n)
 *
 *  n ... kolikata faze se ma vykreslit
 *
 *  vykresli jednu fazi barvy pro vsechny diody
 */

void cykl(int n) {
  for(byte i = 0; i < COLORS; i++) {
    digitalWrite(latchPin[i], LOW);
     byte out = 0;
     for(byte j = 0; j < 8; j++)
       if(n < data[j][i])
         out |= (1 << j);
     shiftOut(dataPin[i], clockPin, MSBFIRST, out);

     out = 0;
     for(byte j = 0; j < 8; j++)
       if(n < data[8+j][i])
         out |= (1 << j);

     shiftOut(dataPin[i], clockPin, MSBFIRST, out);
     digitalWrite(latchPin[i], HIGH);
  }
}
```

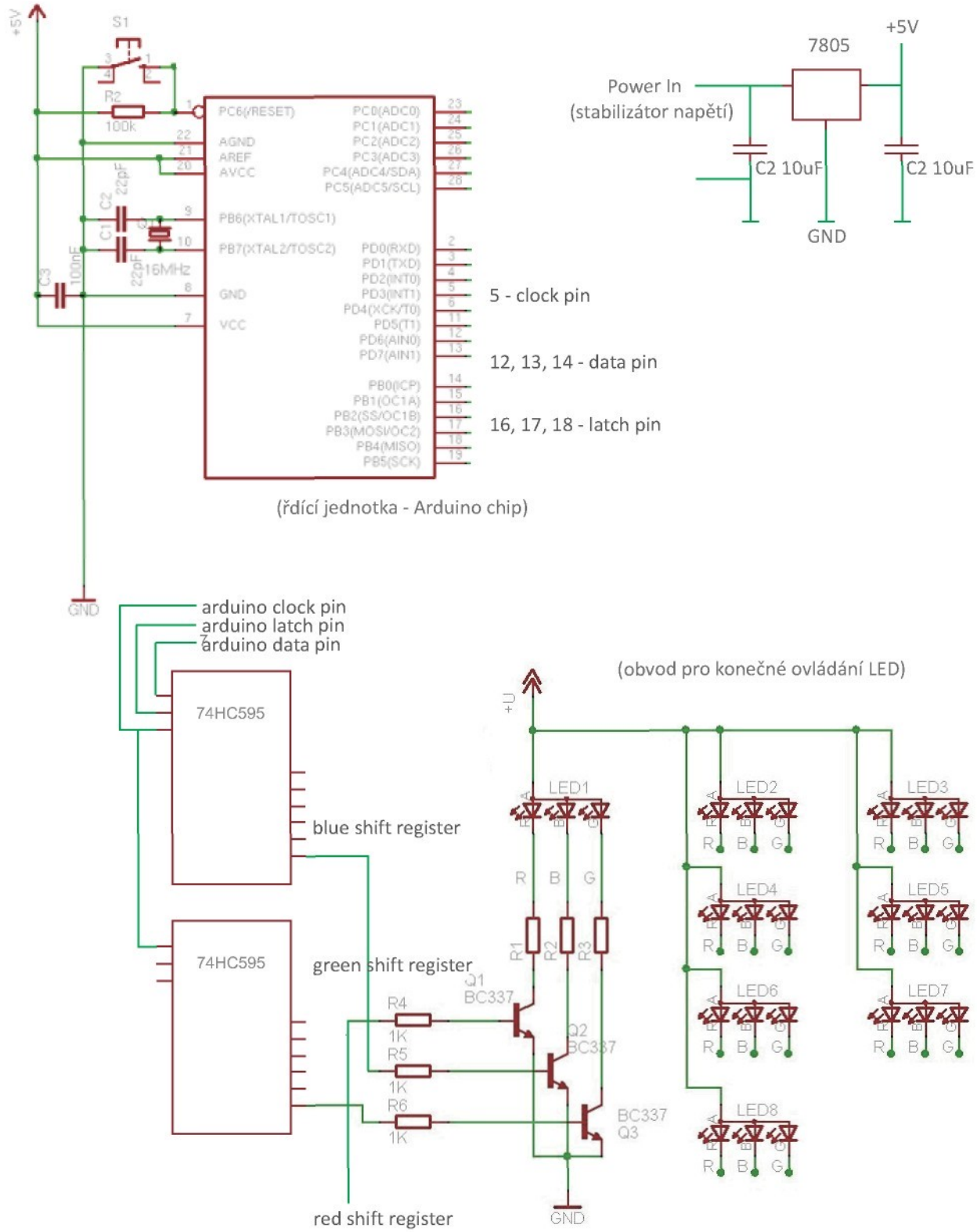
Arduino piny (ATMega 328)



Posuvný registr (Shift register) piny (74HC595)



PINS 1-7, 15	Q0 " Q7	Output Pins
PIN 8	GND Ground,	Vss
PIN 9	Q7"	Serial Out
PIN 10	MR	Master Reclear, active low
PIN 11	SH_CP	Shift register clock pin
PIN 12	ST_CP	Storage register clock pin (latch pin)
PIN 13	OE	Output enable, active low
PIN 14	DS	Serial data input
PIN 16	Vcc	Positive supply voltage


+5V
S1
R2
100k
PC6(/RESET)
AGND
AREF
AVCC
PB6(XTAL1/TOSC1)
PB7(XTAL2/TOSC2)
GND
VCC
C2 22pF
C1 22pF
16MHz
C3 100nF
PC0(ADC0)
PC1(ADC1)
PC2(ADC2)
PC3(ADC3)
PC4(ADC4/SDA)
PC5(ADC5/SCL)
PD0(RXD)
PD1(TXD)
PD2(INT0)
PD3(INT1)
PD4(XCK/T0)
PD5(T1)
PD6(AIN0)
PD7(AIN1)
PB0(ICP)
PB1(OC1A)
PB2(SS/OC1B)
PB3(MOSI/OC2)
PB4(MISO)
PB5(SCK)
5 - clock pin
12, 13, 14 - data pin
16, 17, 18 - latch pin
(řdící jednotka - Arduino chip)
Power In
(stabilizátor napětí)
7805
+5V
C2 10uF
C2 10uF
GND
arduino clock pin
arduino latch pin
arduino data pin
74HC595
blue shift register
74HC595
green shift register
red shift register
(obvod pro konečné ovládání LED)
+U
LED1
LED2
LED3
LED4
LED5
LED6
LED7
LED8
R B G
R1
R2
R3
R4
1K
R5
1K
R6
1K
Q1
BC337
Q2
BC337
BC337
Q3
GND

Konečná práce - instalace těchto několika "interaktivních" sloupů by měla člověka oslovit hlavně vizuálně. Vizuální stránku by měla doplnit skutečnost, že sloupy mají vlastní "život". A nejsou tedy statické, ale různě se variují a to i v závislosti na tom, zda s nimi někdo "komunikuje". Hravá interakce mezi nimi a divákem je pro mne v tomto díle motivací.