Středoškolská technika 2018

Setkání a prezentace prací středoškolských studentů na ČVUT

Scoreboard – výsledková tabule pro volejbal, fotbal, házenou a jiné míčové sporty

Václav Kohout

Gymnázium Kladno,
Nám. Edvarda Beneše 1573
272 01, Kladno

Naše škola je hodně zaměřena na volejbal, který je náplní hodin tělocviku po celou dobu studia. Každoročně se na naší škole pořádá volejbalový turnaj, a hra jako taková se provozuje i v hodinách. Rozhodčí, resp. vyučující tělocviku, musel vždy mít někoho, kdo mu otáčel skóre na mechanickém počítadle, nebo počítal pouze ústně, což není úplně vhodné. Z tohoto důvodu mi přišlo jako dobrý nápad zrealizovat tento projekt.

Rád bych poděkoval RNDr. Mileně Minaříkové, ředitelce školy, která projekt schválila, dále Mgr. Lucii Pelikánové a Mgr. Lence Smyčkové za doporučení zúčastnit se StreTechu.

Počátky vývoje

Nejprve jsem začal vymýšlet softwarové řešení projektu. Pokusy se softwarem jsem prováděl na vývojové desce Arduino Leonardo, ke které mám pro tyto účely zakoupen shield, abych nemusel při každém pokusu sestavovat obvod na nepájivém poli. Na shieldu jsou umístěna 3 tlačítka a čtyřmístný sedmisegmentový displej. Tento displej je řízený dvěma posuvnými registry. Kód po několika pokusech začal fungovat, což mne přesvědčilo o tom, že je projekt proveditelný.

obr.1 – Arduino se shieldem

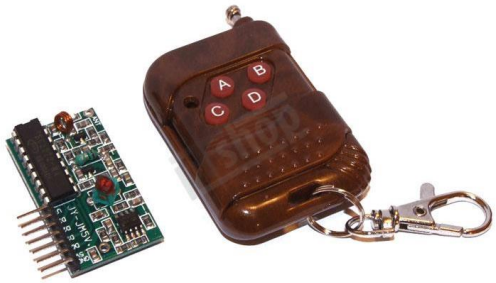
Výběr součástek

K řízení scoreboardu jsem využil procesor z rodiny Atmel ATmega. Jde o procesor, který je používán na deskách Arduino, a lze jej programovat pomocí Arduino IDE. Na trhu jsem nenašel žádný dostatečně velký displej, který by byl pro tyto účely vhodný, proto jsem se rozhodl vyrobit si displej sám. Vyrobil jsem jej ze žlutých LED diod. Diody jsem vybíral podle svítivosti tak, aby byl text co nejlépe čitelný (použité diody mají svítivost 780 mcd při jmenovitém napětí 2,2 V).

Abych nemusel diody spínat jednotlivými tranzistory, sehnal jsem tranzistorové pole (procesor pracuje při 5 V, zatímco diody zapojené v segmentu vyžadují napětí 12 V). K napájení jsem použil dva regulované zdroje (na 5 a 12 V) a tranzistory pro spínání anod (viz obr.3 u výroby plošného spoje). Ostatní součástky neuvádím.

K přičítání, resp. odečítání skóre jsem se rozhodl využít dálkový ovladač. Skládá se z přijímače a přenosného vysílače vybaveného čtyřmi tlačítky. Vysílač pracuje na frekvenci 433 MHz, což zajišťuje větší dosah. Z přijímače vedou čtyři výstupy odpovídající tlačítkům na vysílači. Výhodou tohoto ovládání je, že rozhodčí může stát na opačné straně hřiště. Dalším plusem je to, že displej bude zavěšen tak vysoko, že by nebylo možné jednoduše obsluhovat tlačítka na boxu displeje. Do budoucna se nabízí varianta řešení pomocí IR přijímače, s možným ovládáním pomocí chytrého mobilního telefonu.

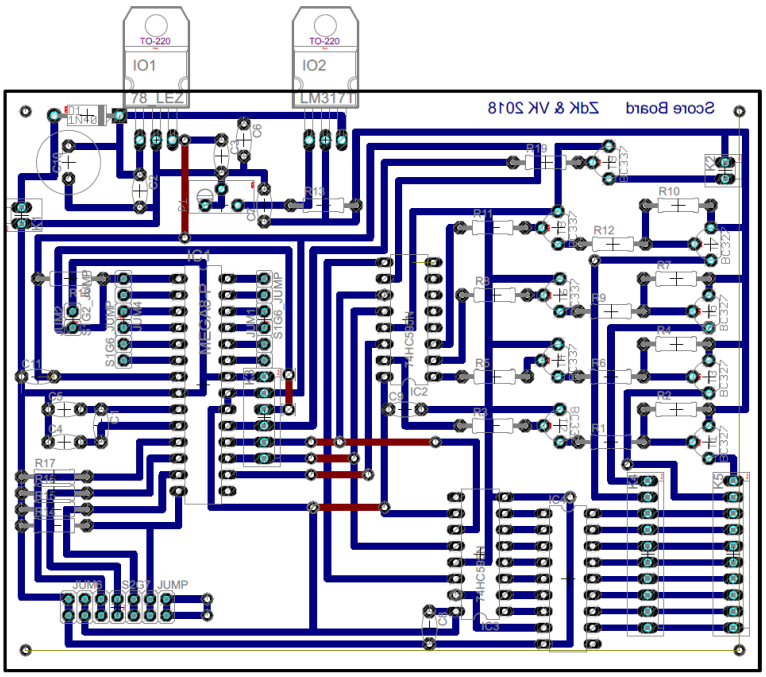
Řízení displeje jsem řešil stejně jako je na shieldu – tj. dvěma posuvnými registry. Toto řešení mi přišlo jako nejrozumnější, a nejjednodušší při programování procesoru.



obr. 2 – dálkový ovladač s přijímačem

Výroba plošného spoje

Vyleptal jsem vlastní desku plošného spoje, návrh je patrný na obrázku. Návrh byl proveden pomocí programu EAGLE.

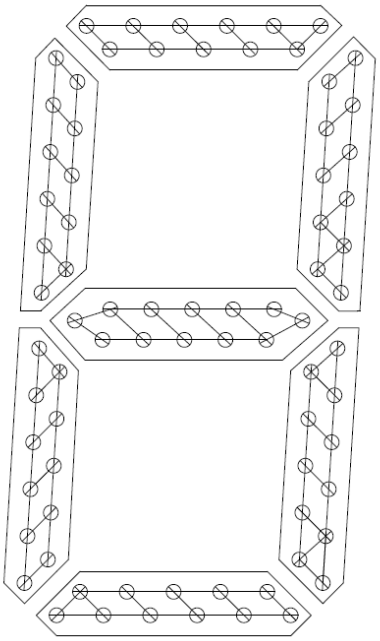


obr. 3 – schéma plošného spoje

obr. 4 – plošný spoj osazený součástkami

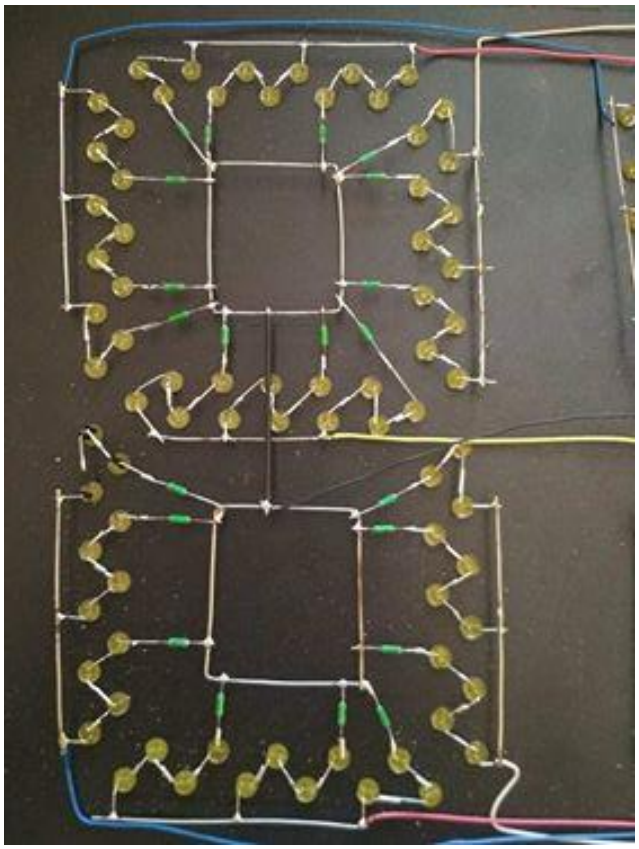
Výroba displeje

Displej je vyroben z 316 diod. Diody jsou usazené do plastové desky o tloušťce 5 mm. Diod je v segmentu 11, jsou tedy zapojeny po 4 s rezistorem o odporu 120Ω (jedna část segmentu jsou 3 diody a 220Ω rezistor). Dvojtečka je tvořena dvěma čtveřicemi diod. Rozměry displeje jsou voleny tak, aby mohl být displej zavěšen v otvoru v žebřinách. Pokud zavěsíme displej do mezery mezi žebřinami, je menší pravděpodobnost, že bude vystaven přímým nárazům balónu.



obr. 5 – schéma rozmístění diod v displeji

obr. 6 – osazování diod do desky



obr.7 – zadní pohled na desku s diodami

Box na displej

Materiál pro výrobu boxu jsem volil tak, aby nebyl příliš těžký, ale zároveň pevný a odolný. Zvolil jsem tmavě hnědou překližku tloušťky 9 mm. Vnější rám má rozměry 27x86 cm, v něm je přesně sednoucí vnitřní rám, na kterém je uchycená deska s diodami (musí být uchycena vevnitř, jinak by totiž konce diod přesahovaly ven z boxu. Čelo displeje je tvořeno plexisklem tloušťky 4 mm z důvodu ochrany před dopady balónu. V zadní stěně boxu je otvor, který je zaslepený deskou z plastu tak. Deska z plastu slouží k tomu, že se běžný uživatel nedostane k plošnému spoji, ale zároveň budeme my mít možnost se k němu v případě potřeby dostat. Z boku displeje se nachází konektor pro připojení 12V adaptéru a vypínač. V zadní desce jsem vyvrtal dva otvory pro uchycení na zeď.

obr. 8 – box na displej

Realizace v prostoru

Školní tělocvična je kulturní památkou, proto je nutné, aby řešení bylo neinvazivní. Dále bylo potřeba počítat s tím, že v místě není rozvod elektřiny, je tedy nutné dovést po stěně kabel z 12V adaptéru. Scoreboard bude uchycen na dvou šroubech tak, aby jej bylo možné sundat (z důvodu konání kulturních akcí v této místnosti).

obr. 9 – fotografie tělocvičny, se znázorněním místa, kam bude scoreboard instalován

obr. 10 – fotografie hotového scoreboardu

Kód

Software pro scoreboard jsem programoval v jazyku Arduino IDE, což je zjednodušené C++. Program má dvě základní funkce – starat se o čtení vstupu z tlačítek a podle toho změny hodnot skóre, a zároveň se stará o zapisování čísel na displej. V každém momentě svítí pouze jedna číslice, proto se musí na první, druhý, třetí, čtvrtý segment vypisovat stále dokola. Vypisuje se to tak rychle, že to lidské oko není schopné zachytit. Mezi jednotlivým vypisováním na segment je pauza 3 milisekundy, protože jinak docházelo k multiplexování na vedlejší segment. Na následujících stránkách je celý kód česky okomentovaný.

```
/*
  © Václav Kohout 2018
*/
#define pinLatch 11
#define pinClk   12
#define pinData  10

#define buttonR1 7
#define buttonL1 5
#define buttonR2 8
#define buttonL2 6

#define ON  LOW
#define OFF HIGH

unsigned long lastSwitchTime = 0;

int left = 0; // proměnné levá a pravá - skóre pro jednotlivé týmy
int right = 0;

int buttonState1;             //definice stavů tlačítek
int lastButtonState1 = LOW;   //- využíváme k tomu, aby při stisku
                              //  a podržení tlačítka nedošlo
                              //  k několikanásobnému přičtení bodů

int buttonState2;
int lastButtonState2 = LOW;
unsigned long lastSwitchTime2 = 0;

int buttonState3;
int lastButtonState3 = LOW;
unsigned long lastSwitchTime3 = 0;

int buttonState4;
int lastButtonState4 = LOW;
unsigned long lastSwitchTime4 = 0;

unsigned long SwitchDelay = 50;

const byte mapNumSeg[] = {0xF1, 0xF2, 0xF4, 0xF8};
                         //maga segmentů pro druhý posuvný registr
const byte mapSeg[]    = {0xBF, 0x86, 0xDB, 0xCF, 0xE6, 0xED, 0xFD, 0x87, 0xFF,
0xEF};
                         //mapa segmentů jednotlivých čísel pro posuvný registr

void printseg(byte seg, byte value) {
  //nastavení funkce printSeg - zapsání čísla na segment
  digitalWrite(pinLatch, LOW);
  shiftOut(pinData, pinClk, MSBFIRST, mapSeg[value]);
  shiftOut(pinData, pinClk, MSBFIRST, mapNumSeg[seg]);
  digitalWrite(pinLatch, HIGH);
}

void clear() {
  //nastavení funkce clear - vynulování skóre
  left = 0;
  right = 0;
}
```

```
void setup () {
  pinMode(pinLatch, OUTPUT);
  pinMode(pinClk, OUTPUT);
  pinMode(pinData, OUTPUT);
  left = 0;
  right = 0;
}

void loop() {
  // v této části programu čteme vstupy z tlačítek
  // a zapisujeme čísla na jednotlivé segmenty
  delay(3);
  // delay je zde z toho důvodu, aby se nám čísla nemultiplexovala
  // na vedlejší číslice displeje
  printseg(1 , left / 10);
  delay(3);
  printseg(0 , left % 10 );
  delay(3);
  printseg(3 , right / 10);
  delay(3);
  printseg(2 , right % 10);
  int reading1 = digitalRead(buttonR1);
  int reading2 = digitalRead(buttonL1);
  int reading3 = digitalRead(buttonR2);
  int reading4 = digitalRead(buttonL2);

  if (reading1 != lastButtonState1) {
    //zde probíhá samotné přičítání podle stavů tlačítek
    lastSwitchTime = millis();
  }

  if ((millis() - lastSwitchTime) > SwitchDelay) {

    if (reading1 != buttonState1) {
      buttonState1 = reading1;

      if (buttonState1 == HIGH) {
        right++;
      }
    }
  }

  if (reading2 != lastButtonState2) {
    lastSwitchTime2 = millis();
  }

  if ((millis() - lastSwitchTime2) > SwitchDelay) {

    if (reading2 != buttonState2) {
      buttonState2 = reading2;

      if (buttonState2 == HIGH) {
        left++;
      }
    }
  }
  if (reading3 != lastButtonState3) {
    lastSwitchTime3 = millis();
  }

  if ((millis() - lastSwitchTime3) > SwitchDelay) {

    if (reading3 != buttonState3) {
      buttonState3 = reading3;
```

```
      if (buttonState3 == HIGH) {
        if (right != 0) {
          right--;
        }

      }
    }
  }
  if ((millis() - lastSwitchTime4) > SwitchDelay) {

    if (reading4 != buttonState4) {
      buttonState4 = reading4;

      if (buttonState4 == HIGH) {
        if (left != 0) {
          left --;

        }

      }
    }
  }
  lastButtonState1 = reading1;
  lastButtonState2 = reading2;
  lastButtonState3 = reading3;
  lastButtonState4 = reading4;
  if ((reading4 == HIGH) && (reading3 == HIGH)) {
    //v případě stisku tlačítek 3 a 4 zároveň dojde k resetování displeje
    clear();
  }
}
```

Závěr

Na začátku tohoto projektu stál vlastně jen nápad, který mi nepřišlo reálné zrealizovat. Na konci stojí hotová výsledková tabule, moje zkušenosti s programováním se o mnohé rozšířily. Projekt je možné dále rozvíjet, neboť plošný spoj je koncipován tak, že procesor je možné kdykoli přeprogramovat, dále je také možné změnit vstupy tlačítek (na plošném spoji je patice, do které je přivedeno +5V, zem a 4 digitální piny, proto je možné do budoucna změnit komunikační modul.