



Středoškolská technika 2013

Setkání a prezentace prací středoškolských studentů na ČVUT

DIGITÁLNĚ ŘÍZENÝ ZDROJ

Radovan Blažek

SPŠE Ječná
Ječná 30, Praha 2

Obsah

Zadání	2
1 Úvod	2
1.1 Základní informace	2
1.2 Princip impulsního zdroje	2
1.3 Druhy spínaných zdrojů	3
1.4 Volba topologie zdroje	3
2 Výkonová část	5
2.1 Zpracování síťového napájení	5
2.2 Spínací tranzistory	5
2.3 Transformátor	6
2.4 Usměrňovač	8
2.5 Filtr	8
3 Řídící část	9
3.1 ATMEGA2560	9
3.2 TL494	9
3.3 IR2101	9
4 Realizace	11
4.1 Deska plošných spojů	11
4.2 Výstupní svorky a konektor	11
5 Software	13
6 Závěr	20
Seznam obrázků	23
Slovník	24
Reference	25

1 Úvod

Tato maturitní práce se zabývá návrhem a realizací spínaného zdroje řízeného mikrokontrolérem. Digitální řízení zdroje umožňuje celou řadu využití takového zdroje. Kromě základního použití jako laboratorní zdroj a zdroj pro napájení nejrůznějších zařízení, může sloužit i k dalším procesům jako například nabíjení a to článků akumulátorů jakéhokoliv chemického složení. Rozdíl mezi nabíjením jednotlivých typů baterií je pouze v algoritmu. Mikrokontrolér nám tedy umožní vykonávat tyto algoritmy, podle našeho výběru v uživatelském prostředí. Nabíjení, ale není jediný možný způsob využití možností digitálního řízení. Další se skrývají v přesné regulaci například teploty pod infra páječkou nebo teploty v teráriu¹, regulace proudu při galvanickém pokovování atd. Největší síla je v naprosté univerzálnosti - takovýto zdroj dokáže nahradit jakýkoliv specializovaný zdroj stejnosměrného napětí, pokud na to stačí výkonově.

1.1 Základní informace

Spínaný zdroj je přístroj, který slouží k napájení elektronických zařízení. Od lineárních zdrojů se liší tím, že pracuje i přímo se síťovým napětím. Existují spínané zdroje s pracovním kmitočtem sítě a s kmitočtem vyšším než síťovým. Tato práce se bude zabývat pouze druhým zmíněným typem. Oproti lineárním zdrojům má spínaný zdroj s kmitočtem vyšším než síťovým mnoho výhod. K těm hlavním patří:

- vyšší účinnost
- menší transformátor než u zdroje lineárního, při stejném výkonu

Z toho vyplývají sekundární výhody jako:

- nižší hmotnost
- nižší cena za stejný výkon
- menší celkové rozměry

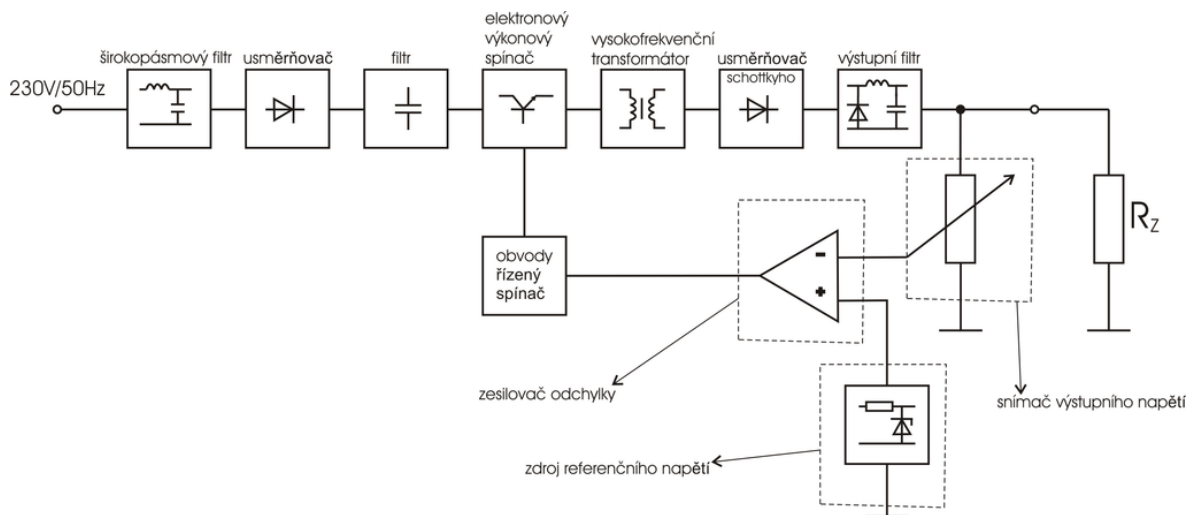
Samozřejmě se neobejde ani bez nevýhod:

- složitější zapojení
- rušení na výstupu vzniklé spínáním

Kvůli rušení se spínané zdroje nehodí k audio zesilovačům, některým měřicím/laboratorním přístrojům a dalším zařízením citlivým na toto rušení. Jejich výhody, ale pro běžné aplikace převyšují a proto se dnes nachází v naprosté většině zařízení.

1.2 Princip impulsního zdroje

Spínaný zdroj nejdříve usměrní a vyfiltruje síťové napětí. Za filtrem se pomocí tranzistorů pošle impulsy do impulsního transformátoru, ten napětí o něco sníží. Impulsy se usměrní a vyfiltrují výstupním filtrem. Regulace spočívá v tom, že zpětná vazba z výstupního napětí nebo proudu, podle toho co chceme regulovat, ovlivňuje šířku impulsů jdoucích do transformátoru. Čím jsou impulsy širší tím se přenesou více energie a napětí resp. proud na výstupu je větší.



Obrázek 1: Blokové schéma spínaného zdroje [8]

1.3 Druhy spínaných zdrojů

Spínaných zdrojů je několik typů. Existují zdroje bez indukčností, takzvané nábojové pumpy a s indukčnostmi, které mohou být s galvanickým oddělením (s transformátorem) nebo bez. Zdroje s galvanickým oddělením se používají např. kvůli bezpečnosti, když je zdroj napájený ze síťového napětí. Zdroje bez galvanického oddělení se používají u zdrojů napájených např. z baterií, protože vstupní napětí není nebezpečné a není většinou ani potřeba velké změny napětí, pro kterou je transformátor jinak vhodný.

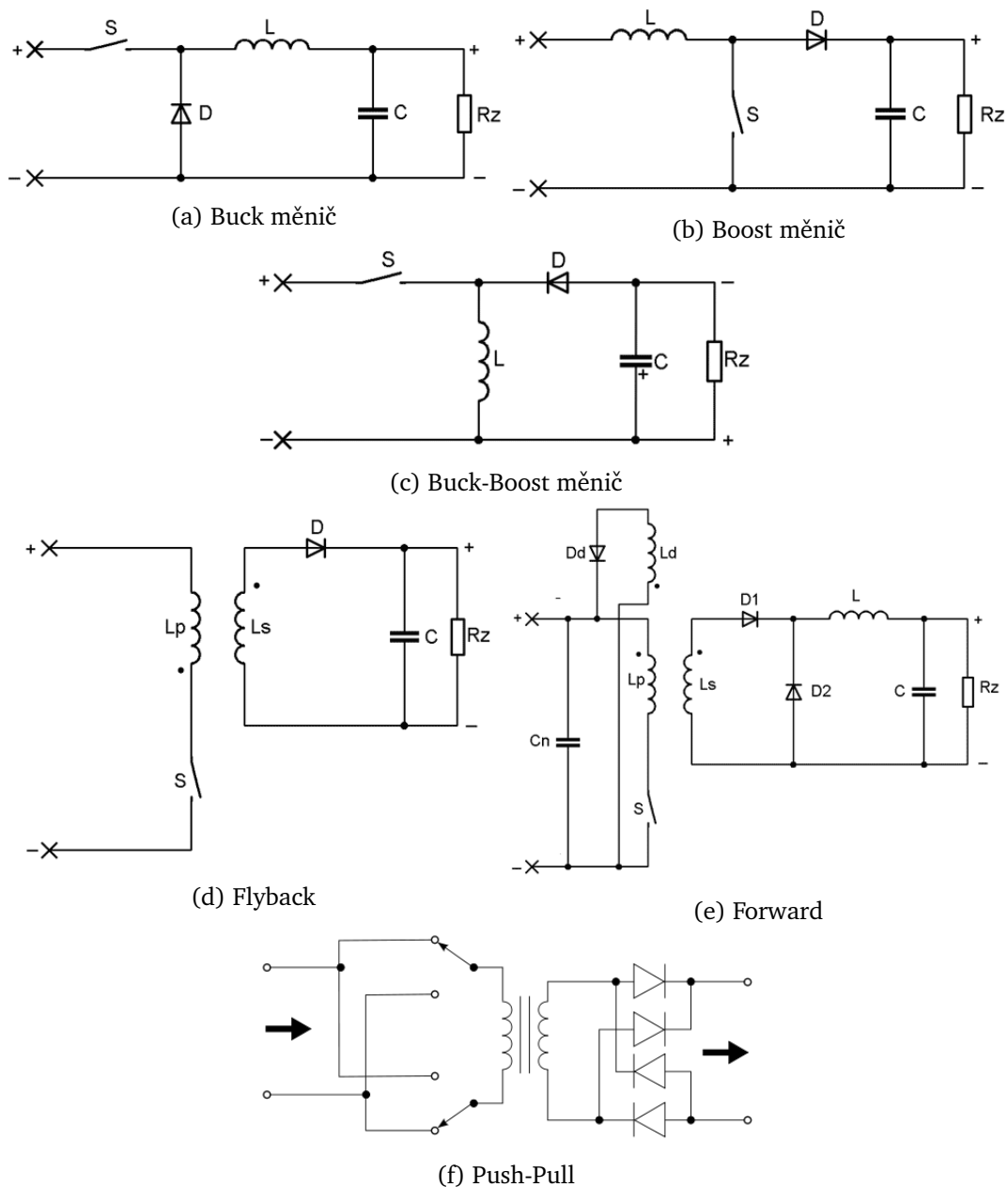
1.4 Volba topologie zdroje

Spínané impulsní zdroje lze rozdělit podle topologií například takto:

- Bez galvanického oddělení
 - Snižovač - Buck
 - Zvyšovač - Boost
 - Invertor - Buck-Boost
- S galvanickým oddělením
 - Blokující - Flyback
 - Propustný - Forward
 - * S odbočkou na primárním vinutí - Push-Pull
 - * Polomost - Half Bridge
 - * Plný můstek - Full Bridge

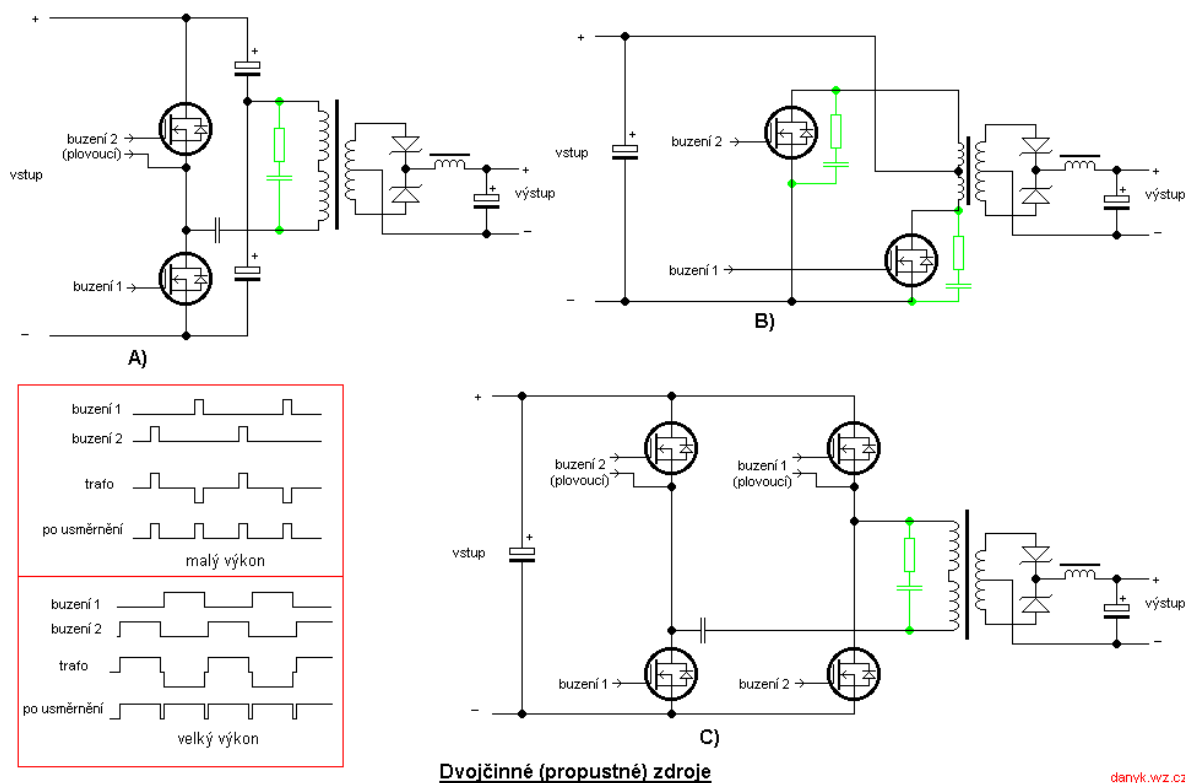
Topologie se většinou² nazývají stejně, ale různí autoři kategorie rozdělují nebo slučují, protože rozdělení není úplně jednoznačné a některé topologie se dost podobají. Výběr topologie zdroje samozřejmě závisí na požadovaných parametrech hotového zdroje. Co se týče parameterů, liší se hlavně ve výkonu a složitosti. Zvolil jsem topologii, která poskytuje nejlepší poměr

výkon/složitost - propustný dvojitý zdroj, verze s polomostem viz. obrázek 3. Polomost střídavě připojuje jednu stranu primární cívky transformátoru ke kladnému a zápornému napětí pomocí dvou tranzistorů. Druhá strana je připojena mezi dva kondenzátory, které tu vytváří polovinu vstupního napětí.



Obrázek 2: Topologie DC/DC měničů a AC/DC zdrojů [1]

²Například pojmem Push-Pull se označuje skupina topologií, která použít proud do primární cívky transformátoru jedním i druhým směrem, ale někdy se používá jen pro topologii s odbočkou na primární cívce.



Obrázek 3: Propustné dvojčinné - Forward Push-Pull [2]

2 Výkonová část

Popis jednotlivých komponent.

2.1 Zpracování síťového napájení

Za přívodem síťového napětí následuje filtr na odrušení sítě od spínání tranzistorů. Dále pojistka a NTC termistor omezující proudový ráz při zapojení. Nakonec usměrňovací můstek a dva kondenzátory. Kondenzátory mají dvě funkce, filtrují usměrňované napětí a vytvářejí mezi sebou polovinu napájecího napětí.

2.2 Spínací tranzistory

Tranzistory jsou použity unipolární, typ **MOSFET**. Pro tuto aplikaci jsou podstatné hlavně tyto parametry:

- maximální dovolené napětí na tranzistoru U_{DS}
- maximální dovolený proud tranzistorem I_D
- potřebný náboj hradla Gate Q_G
- odpor v sepnutém stavu R_{DS}

Přičemž první dva nesmí klesnout pod určitou hranici a druhé dva pouze zlepšují účinnost zdroje. Na náboji hradla přímo závisí jak dlouho se bude tranzistor otevírat a zavírat. Čím déle tento děj probíhá, tím větší je ztrátový výkon na tranzistoru. U spínaného zdroje se tento děj neustále opakuje, proto je tento parametr důležitý z hlediska účinnosti a chlazení zdroje. Na odporu R_{DS} zase závisí ztráty v již sepnutém stavu.

Z dostupných se zdály nejvíce vyhovující STP9NK50Z od firmy ST Microelectronics.

Symbol	Popis	Hodnota
U_{DS}	Maximální napětí mezi Drain a Source	500 V
I_D	Maximální proud tekoucí přes Drain	7.2 A
$R_{DS(on)}$	Odpor v sepnutém stavu	0.72 Ω

Tabulka 1: Parametry STP9NK50Z

2.3 Transformátor

Jedna z nejdůležitějších částí na zdroji je transformátor. V tomto případě transformátor impulsní. Transformátor musí mít správné parametry pro aplikaci pro kterou je použit. U impulsních zdrojů se nejčastěji používá jádro tvaru EI nebo EE.

Vyrábí se z feritu, což je keramický oxid, který se používá v jádrech transformátorů kvůli své vysoké permeabilitě a nízké elektrické vodivosti. Čím je permeabilita jádra vyšší tím více jsou do něj vtahovány pomyslné magnetické siločáry - magnetické pole vytvořené primární cívkou transformátoru se soustřeďuje v jádře a tím pádem i silněji působí na sekundární cívku, která je na tomto jádře také namotána. Takže se přenesou více energie. Při změnách magnetického pole se, ale indukuje napětí nejen v cívkách na jádře, ale i v jádře samotném což způsobí vznik výřivých proudů. Tento jev působí tím více čím je vyšší budící frekvence a snižuje účinnost, jelikož se energie ztrácí v jádře, kde se pouze přemění v teplo. Proto je výhodné, pokud má jádro nízkou vodivost, která proudy v jádře omezuje.

Použil jsem transformátor ze zdroje ATX z PC. Tento transformátor je určený pro spínací frekvenci okolo 30 kHz s topologií polomost s napájením ze sítě. Tuto konfiguraci jsem zachoval, proto jsem primární vinutí neměnil. Sekundárních vinutí bylo několik, ty bylo potřeba odstranit a nahradit pouze jedním s vyšším počtem závitů, aby zdroj mohl dodávat i větší napětí. Počet závitů jsem zvětšil tak, aby výstupní napětí při shodném proudu jaký byl u 12 V větve ve zdroji ATX bylo přibližně 20 V.

Transformátor se musel vyvařit ve vodě, aby šel rozebrat. Viz obrázek 5. Primární vinutí od sekundárního je odizolované vrstvou polyimidové pásky a odstíněné od elektrostatického rušení měděným plíškem. Viz obrázek 6. Sekundární vinutí je vinuto bifilární, protože u použitých spínacích frekvencí se už projevuje povrchový jev tzv. skin efekt, kdy se proud vytlačuje k povrchu vodiče a neteče jeho středem. Bifilární vinutí zvětšuje povrch sekundárního vinutí.



Obrázek 4: Feritová jádra



Obrázek 5: Vaření transformátoru



(a) Převinutí transformátoru 2 (b) Převinutí transformátoru 2 (c) Převinutí transformátoru 3

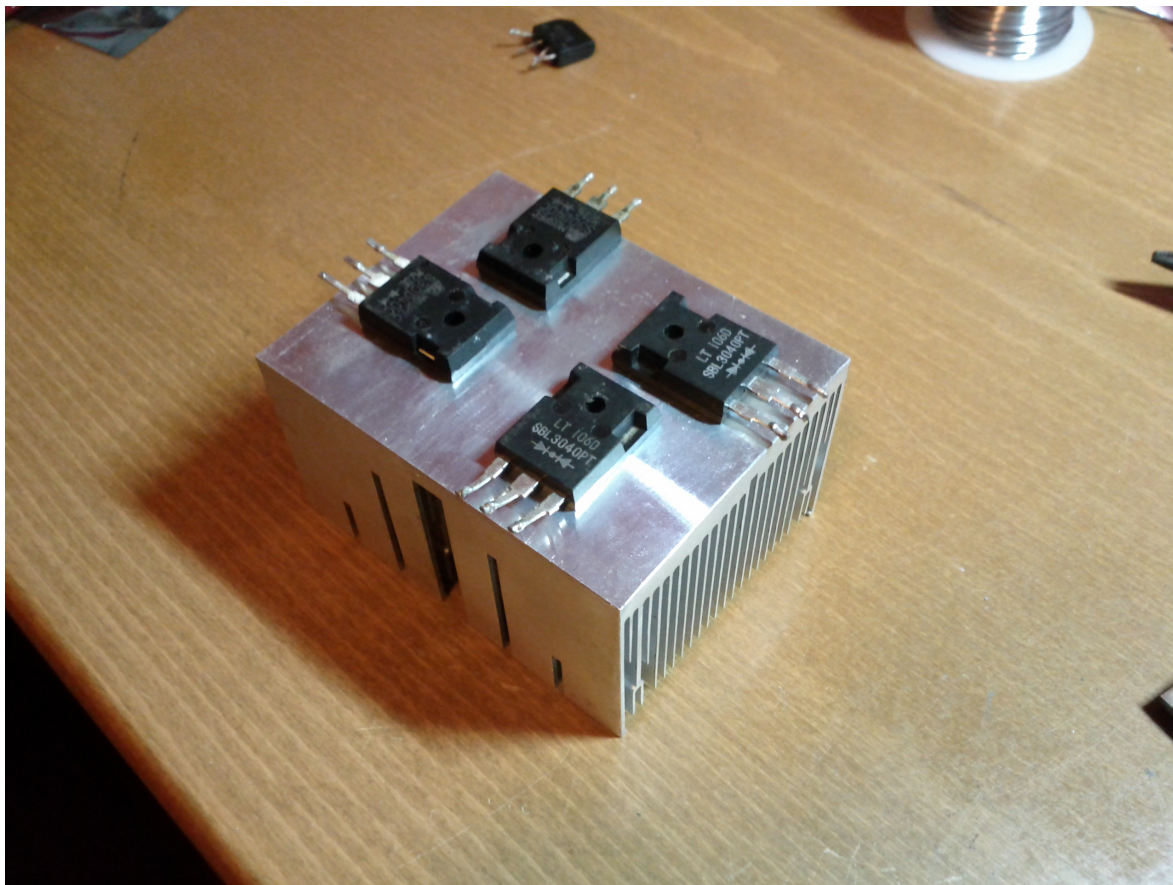


(d) Převinutí transformátoru 4 (e) Převinutí transformátoru 5 (f) Převinutí transformátoru 6

Obrázek 6: Převinutí transformátoru

2.4 Usměrňovač

U většiny spínaných zdrojů s topologií polomost je usměrňovač s odbočkou na sekundárním vinutí transformátoru, protože usměrňuje dvoucestně, ale zároveň je ztráta jen na jedné diodě. To však vyžaduje již zmíněnou odbočku a dvojnásobek sekundárního vinutí. Bohužel dvojnásobek vinutí na sekundární straně by se na použitý transformátor nevešel, takže jsem byl nucen použít Graetzův můstek.



Obrázek 7: Diody s chladičem

2.5 Filtr

Filtr je LC s kapacitou 4000 μF a indukčností 1 μH . Kondenzátory jsou použity s nízkým náhradním sériovým odporem tzv. Low ESR. Tím pádem jsou schopné dávat větší proudy než obyčejné kondenzátory. Frekvence zlomu:

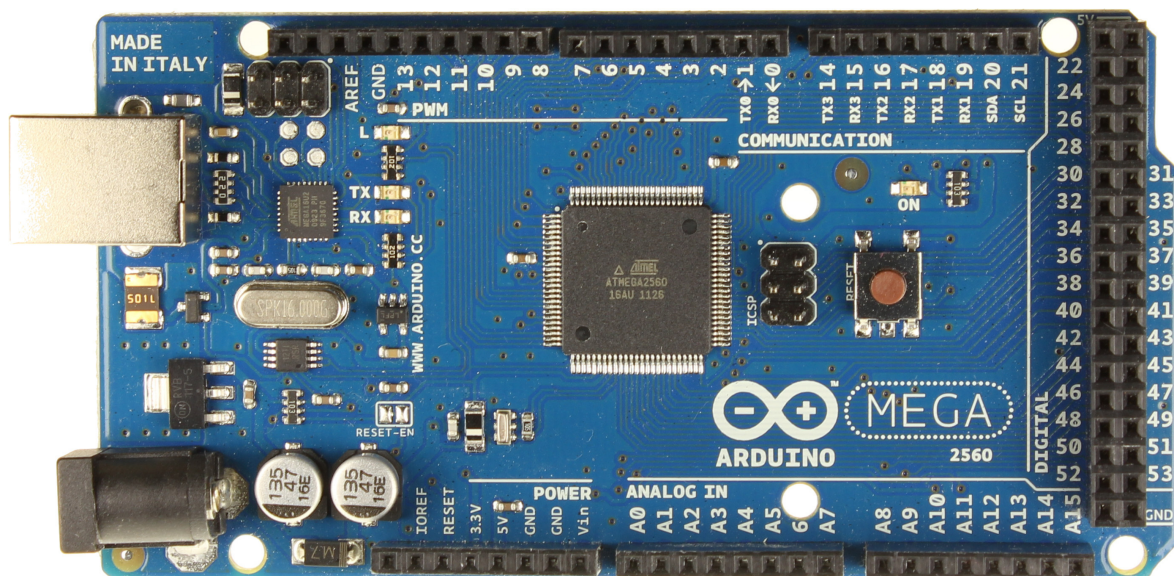
$$f_c = \frac{1}{2\pi\sqrt{LC}}$$
$$f_c = \frac{1}{2\pi\sqrt{1 \cdot 10^{-6} \cdot 4000 \cdot 10^{-6}}} \doteq 2516 \text{ Hz}$$

3 Řídící část

Mikrokontrolér nastavuje výstupní napětí pomocí DAC, které je připojené k řídicímu obvodu TL494.

3.1 ATMEGA2560

Hlavní řídicí jednotka je mikrokontrolér od firmy Atmel s jádrem AVR typ ATMEGA2560-16AU. Mikrokontroléry AVR jsou jedny z nejpoužívanějších osmibitových mikrokontrolérů na světě. ATMEGA2560 má 256 kB paměti typu flash pro program, 8 kB SRAM, maximálně 86 GPIO pinů, SPI, TWI³ a dokáže pracovat maximálně na frekvenci 16 Mhz s výkonem 16 MIPS. Atmega 2560 byla zvolena, dostatečnému počtu GPIO pinů, velikosti paměti, dobré dostupnosti a nízké ceně. Ve zdroji je použita vývojová deska Arduino Mega 2560 s tímto mikrokontrolérem, která disponuje také převodníkem USB na TTL sériovou linku, napěťovým stabilizátorem, 16Mhz krystalem a vyvedenými piny mikrokontroléru na dutinkové lišty.



Obrázek 8: Arduino Mega 2560. Zdroj: <http://arduino.cc/>

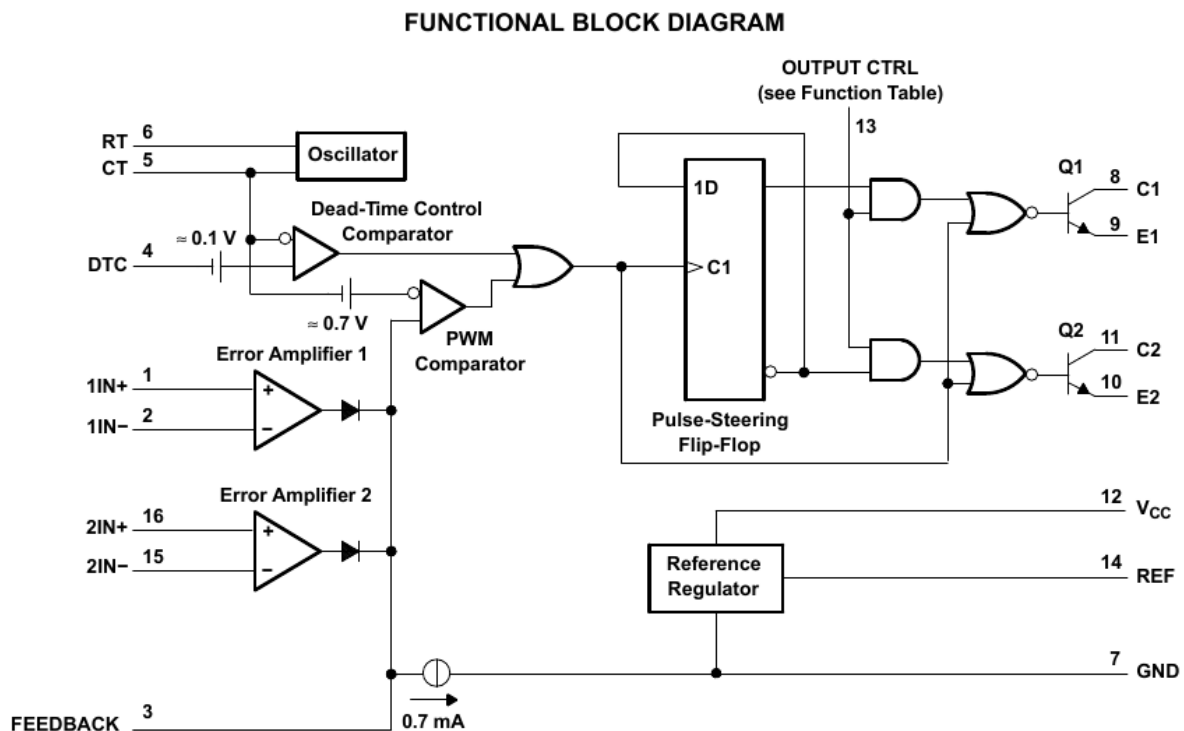
3.2 TL494

Samotnou regulaci - spínání tranzistorů pomocí PWM má na starosti řídicí obvod pro spínané zdroje TL494 od firmy Texas Instruments. Blokové schéma viz. obrázek 9. Tento obvod se vyskytuje téměř ve všech počítačových zdrojích a používá se hojně i v dalších zdrojích s topologií polomost a nejen s ní.

3.3 IR2101

IR2101 je takzvaný high and low side driver pro řízení tranzistorů MOSFET a IGBT v polo-mostu. Polomost tvoří dva tranzistory nad sebou. V našem případě to jsou unipolární tranzis-

³TWI je označení firmy Atmel pro I²C



Obrázek 9: Blokové schéma TL494. Zdroj: <http://ti.com/>

tory MOSFET s kanálem vodivosti N. Tyto tranzistory se otevřou pokud napětí mezi elektrodami Gate a Source U_{GS} vzroste nad určitou hranici. Vybudit tranzistor, který má elektrodu Source na zemi, takzvaný low side tranzistor, není problém. Problém nastává v případě druhého tranzistoru, jenž má elektrodu Source na zátěži. U tohoto tranzistoru jsou na elektrodě Source jednotky stovek voltů, připojených z druhé strany zátěže. Budit tranzistor takto vysokým napětím není jednoduché a nemusí být ani bezpečné, proto se používá plovoucí buzení. To se vytvoří tak, že se budící napětí vztáhne k elektrodě Source a ne k zemi. Nejčastěji se k tomu využívá budícího transformátoru nebo právě integrovaného obvodu jako je IR2101 zahrnující v sobě mikrotransformátory, které vytvářejí zmíněné plovoucí buzení, anglicky floating channel.

Budící IO

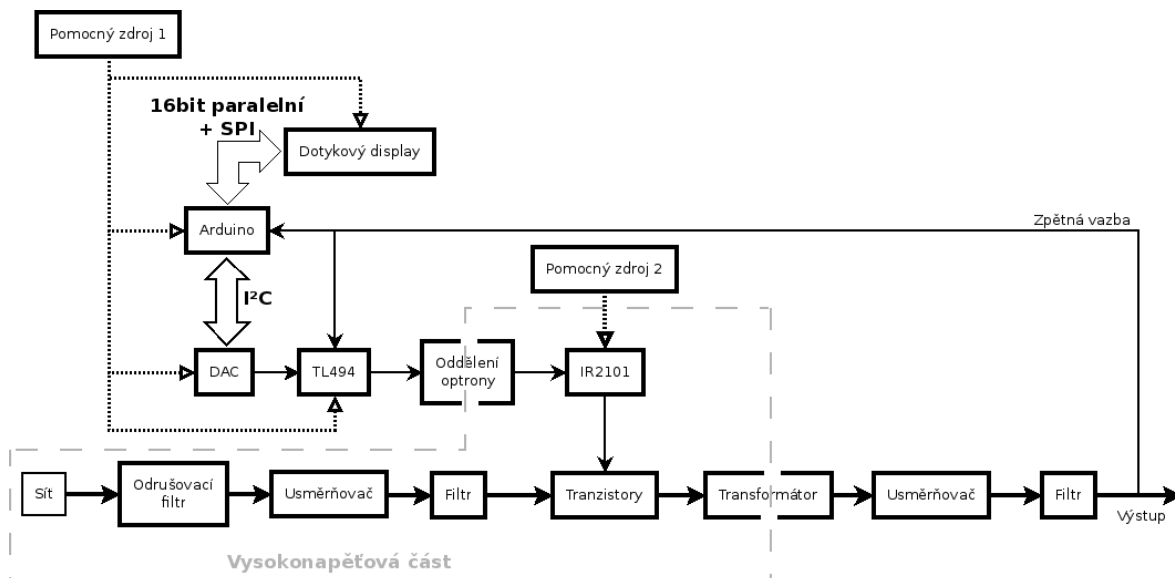
- neposkytuje galvanické oddělení budících signálů
- nepotřebuje žádné další obvody nebo součástky⁴
- přímo uzpůsoben k buzení MOSFET nebo IGBT tranzistorů

Budící transformátor

- poskytuje galvanické oddělení budících signálů
- větší
- dražší
- je nutné ho doplnit o další zapojení

4 Realizace

Princip fungování lze pochopit z blokového schématu na obrázku 10



Obrázek 10: Blokové schéma

4.1 Deska plošných spojů

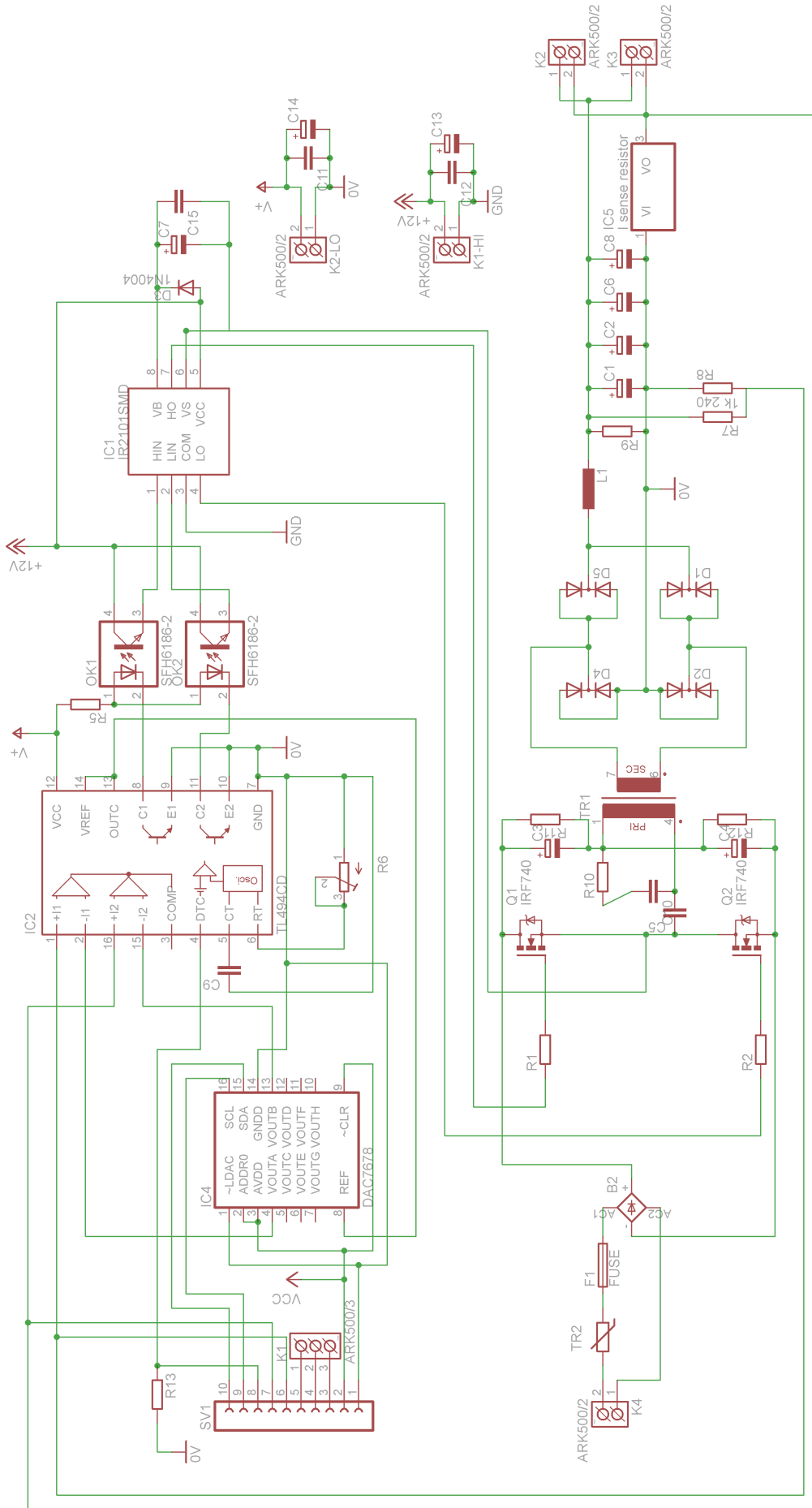
Deska plošných spojů je navrhnutá v programu Eagle. Je oboustranná a je navrhnutá tak, aby se přes ní přímo přišrouboval chladič na diody tvořící výstupní usměrňovač.

4.2 Výstupní svorky a konektor

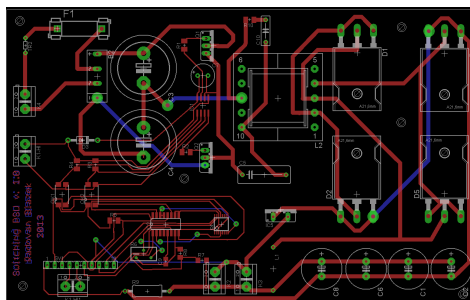
Jelikož má mít zdroj i relativně větší proudový výstup, jsou kladeny i nároky na výstupní konektory. Zdroj má tři možnosti připojení k napájenému zařízení.

- zdířky na banánky
- přišroubování drátu nebo licny ke šroubovací zdířce
- konektor EC5 viz. obrázek 13

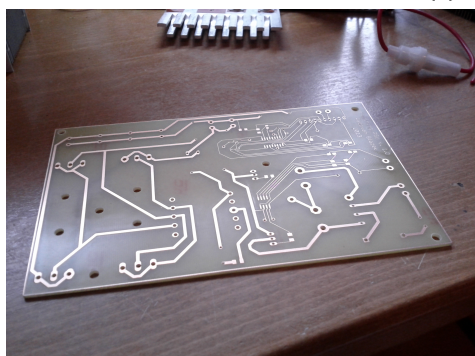
Konektor EC5 se používá k připojování lithiových akumulátorů v RC modelech. Zvládá nepřetržitě zatížení proudem 100 A. Skládá se z dvojice precizních pozlacených banánek a zdířek v plastovém krytu zajišťující připojení vždy se správnou polaritou.



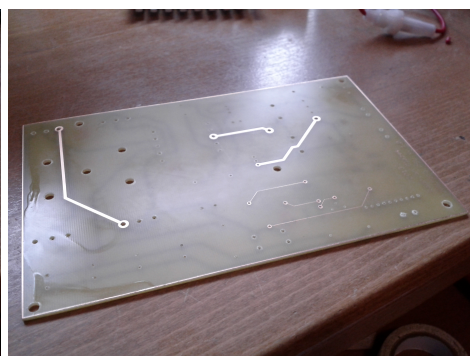
Obrázek 11: Schema



(a) Návrh

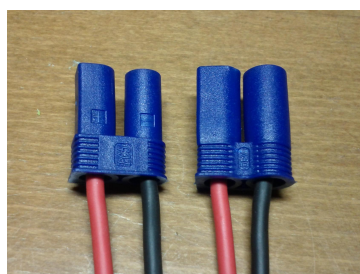


(b) Spodní strana



(c) Vrchní strana

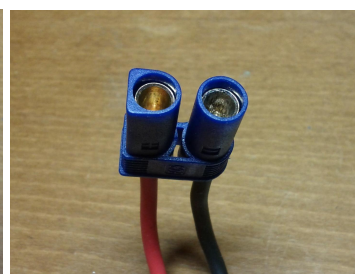
Obrázek 12: Deska plošných spojů



(a) EC5



(b) Samec EC5

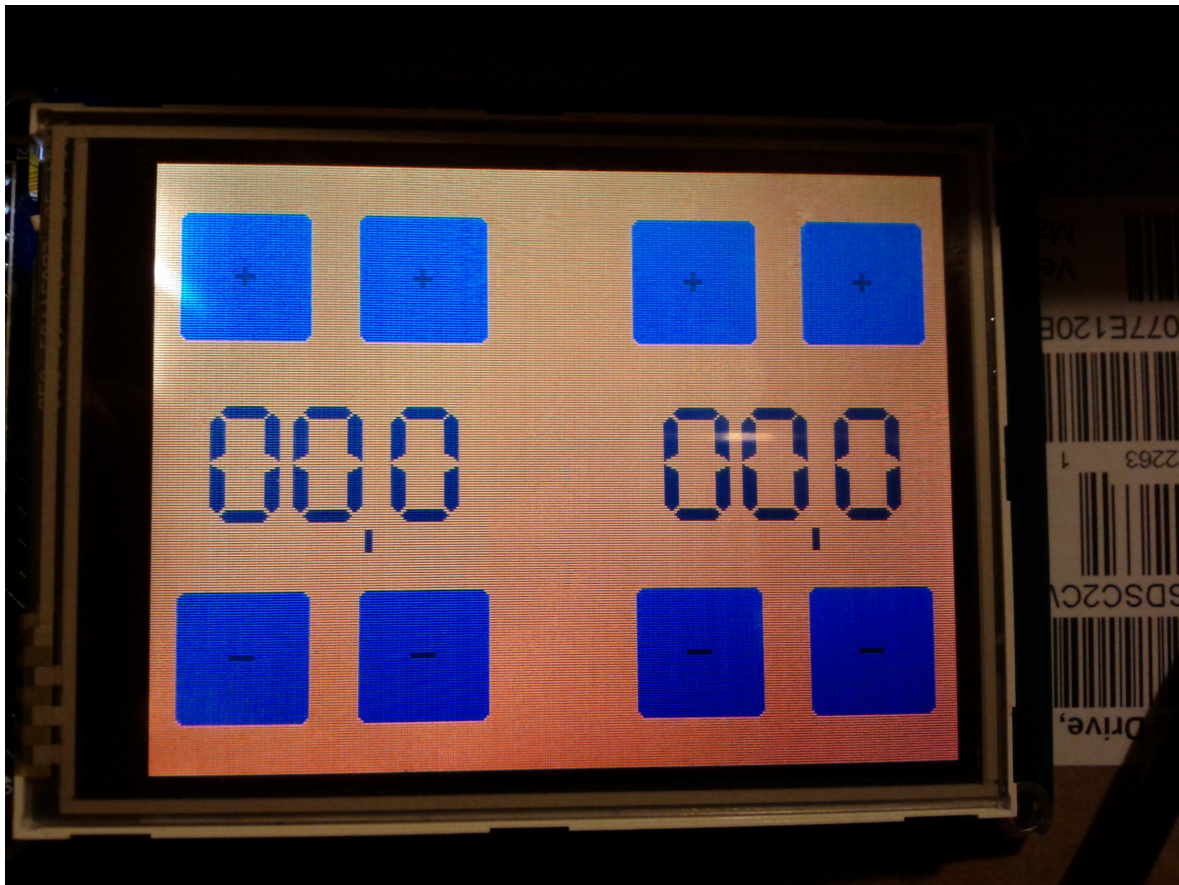


(c) Samice EC5

Obrázek 13: Konektor EC5

5 Software

Program je napsaný v jazyce C++. Jsou použity knihovny pro ovládání displaye, jeho dotykové vrstvy a sběrnice I2C. Program po inicializaci vykreslí na obrazovku ovládací rozhraní napětí a proudu a tlačítko "Menu". Pokud uživatel pomocí ovládacích prvků na displayi změní hodnotu napětí nebo proudu, hodnota se přepočítá, program nastaví napětí DAC a překreslí zobrazenou hodnotu na displayi.



(a) Základní obrazovka

Obrázek 14: Vzhled GUI

Následuje zdrojový kód:

```
#include <UTFT.h>
#include <UTouch.h>
#include <Wire.h>

#include <string.h>

// Declare which fonts we will be using
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
extern uint8_t SevenSegNumFont[];

// Uncomment the next line for Arduino Mega
UTFT myGLCD(ITDB32S,38,39,40,41);
UTouch myTouch(6,5,4,3,2);

int DAC_set(uint8_t channel, uint16_t value, uint8_t valuefrac)
{
```

```

uint8_t MSB,LSB;

value=(819*value)+(82*valuefrac);           //convert voltage to 12bit DAC value (1V =
LSB=value;                                  //split DAC value into MSB and LSB
MSB=value>>8;
Wire.beginTransmission(0x4A);               //send DAC7678 address 0x4A
Wire.write(channel);                         //select channel
Wire.write(MSB);                             //send MSB
Wire.write(LSB);                             //send LSB
int status = Wire.endTransmission();
}

class button
{
public:
    int x,y,sizeX,sizeY,value;
    char colorR,colorG,colorB;
    char * label;

    button();
    ~button();

    void show();
    bool touched(long _x, long _y);
    void handleTouch();
};

class valueControls
{
public:
    int x,y;
    unsigned char value,valuefrac;
    button plusIntVoltButton;
    button minusIntVoltButton;
    button plusFracVoltButton;
    button minusFracVoltButton;

    valueControls();
    ~valueControls();

    void show(int _x,int _y);
    bool handleTouch(long _x,long _y);
    void redrawValue();
};

class menu
{

```



```

public:
    button menuButton;

    menu();
    ~menu();

    void handleTouch(long _x, long _y);
    void showMenu();
    void showMenuButton();
};

menu::menu()
{
    menuButton.x = 130;
    menuButton.y = 100;
    menuButton.sizeX = 65;
    menuButton.label = "menu";
}

void menu::showMenu()
{

}

void menu::showMenuButton()
{
    menuButton.show();
}

void menu::handleTouch(long _x, long _y)
{
    if(menuButton.touched(_x, _y))
        showMenu();
}

valueControls::valueControls()
{
    x=0;
    y=0;
    value=0;
    valuefrac=0;
}

void valueControls::show(int _x, int _y)
{
    x=_x;    //190
    y=_y;    //20
}

```

```

plusIntVoltButton.x = x; //button creation and draw
plusIntVoltButton.y = y;
plusIntVoltButton.label = "+";
plusIntVoltButton.show();

minusIntVoltButton.x = x;
minusIntVoltButton.y = y+150;
minusIntVoltButton.label = "-";
minusIntVoltButton.show();

plusFracVoltButton.x = x+70;
plusFracVoltButton.y = y;
plusFracVoltButton.label = "+";
plusFracVoltButton.show();

minusFracVoltButton.x = x+70;
minusFracVoltButton.y = y+150;
minusFracVoltButton.label = "-";
minusFracVoltButton.show();

myGLCD.setColor(2, 81, 103);
myGLCD.setBackgroundColor(255, 129, 0);
myGLCD.setFont(BigFont);
myGLCD.print("|", x+65, y+120);

redrawValue();
}

bool valueControls::handleTouch(long _x,long _y) //returns true if value changed
{
    bool valueChanged=true;
    if(plusIntVoltButton.touched(_x,_y))
    {
        value++;
    }else if(minusIntVoltButton.touched(_x,_y)) //handling of button actions
    {
        value--;
    }else if(plusFracVoltButton.touched(_x,_y))
    {
        valuefrac++;
    }else if(minusFracVoltButton.touched(_x,_y))
    {
        valuefrac--;
    }else

```

```

        valueChanged=false;

    if(valuefrac==10)
    {
        valuefrac=0;
        value++;
    }
    else if(valuefrac==255)
    {
        valuefrac=9;
        value--;
    }
    if(value==21) //limitation of value range
        value=0;
    else if(value==255)
        value=20;

    return valueChanged;
}

void valueControls::redrawValue()
{
    myGLCD.setColor(255, 129, 0); //value background redraw
    myGLCD.fillRoundRect(x+10, y+75, x+10+32*3, y+75+50);

    myGLCD.setColor(2, 81, 103);
    myGLCD.setBackColor(255, 129, 0);
    myGLCD.setFont(SevenSegNumFont); //value redraw
    myGLCD.printNumI(value, x+10, y+75, 2, '0');
    myGLCD.printNumI(valuefrac, x+80, y+75);
}

button::button()
{
    value=0;
    x,y=0;
    sizeX=50;
    sizeY=50;
    colorR=5;
    colorG=125;
    colorB=159;
    label="label";
}

void button::show()
{
    myGLCD.setColor(colorR, colorG, colorB);

```

```

myGLCD.fillRoundRect(x, y, x+sizeX, y+sizeY);
myGLCD.setColor(2, 81, 103);
myGLCD.setBackColor(colorR, colorG, colorB);
myGLCD.setFont(BigFont);
myGLCD.print(label, x-(strlen(label)*8)+(sizeX/2), y-8+(sizeY/2));
}

bool button::touched(long _x, long _y)
{
    return(_x>=x && _y>=y && _x<=x+sizeX && _y<=y+sizeY);
}

void setup()
{
    Wire.begin();

    myGLCD.InitLCD();
    myGLCD.setFont(BigFont);

    myTouch.InitTouch();
    myTouch.setPrecision(PREC_LOW);

    // TCCR0B &= ~7; //erase first 3 bits - prescaler settings
    TCCR0B |= 1; //set PWM frequency on pins 13 and 4 to 62kHz

    pinMode(13,OUTPUT);
    analogWrite(13,0);
}

void loop()
{
    long x, y; //for touch

    myGLCD.clrScr();
    myGLCD.fillScr(255, 129, 0);

    valueControls voltageControls;
    valueControls currentControls;

    voltageControls.show(190,20);
    currentControls.show(10,20);

    menu fcnMenu;
    fcnMenu.showMenuButton();

    while(1)

```

```

{

while(myTouch.dataAvailable() == false);           //wait for touch

myTouch.read();
x = myTouch.getX();
y = myTouch.getY();

if(voltageControls.handleTouch(x,y))
{
DAC_set(0, voltageControls.value, voltageControls.valuefrac);
voltageControls.redrawValue();
}
if(currentControls.handleTouch(x,y))
{
DAC_set(1, currentControls.value, currentControls.valuefrac);
currentControls.redrawValue();
}

while(myTouch.dataAvailable() == true);           //wait for release
delay(100);                                       //eliminate hazards
}

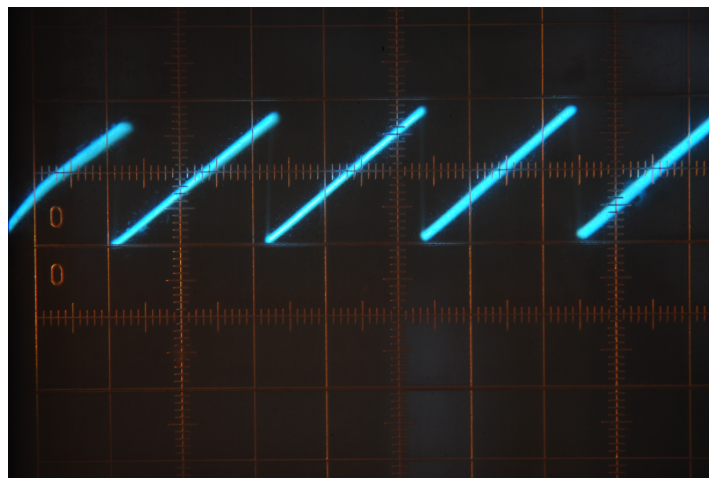
}

```

6 Závěr

Stavba zdroje se zdařila a potvrdila, že takto navrhnutý zdroj může fungovat. Vše bylo otestováno kvůli bezpečnosti zatím na nízkém napájecím napětí. Na obrázku 15 lze vidět zaznamenané průběhy naměřené na hotovém zdroji.

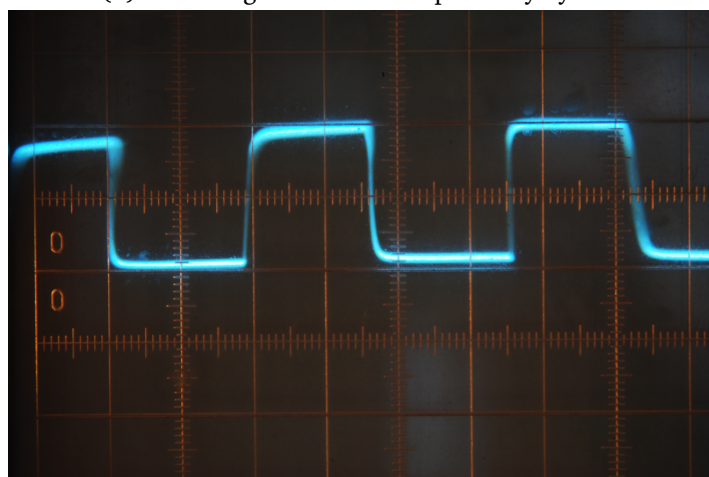
Spínaný zdroj řízený mikrokontrolérem pomocí DAC je realizovatelný.



(a) Průběh napětí na RC oscilátoru TL494.

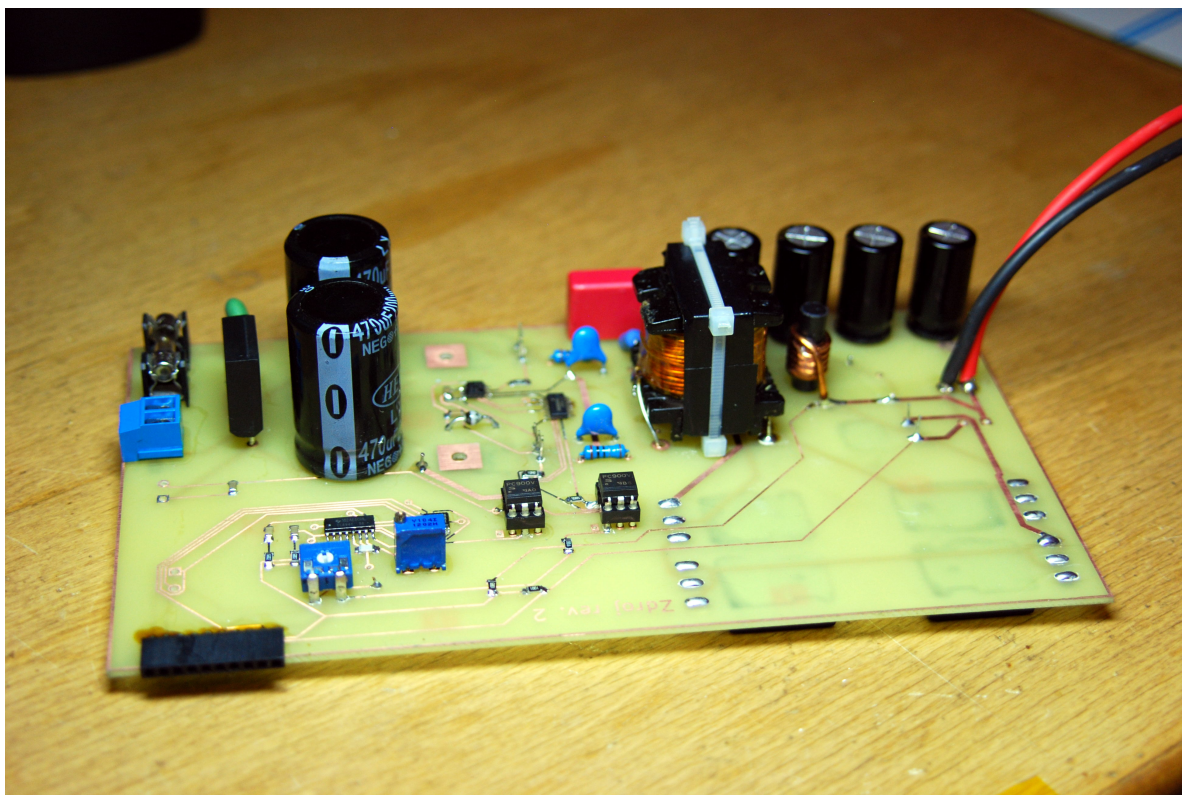


(b) Budící signál tranzistoru pro malý výkon.



(c) Budící signál tranzistoru pro velký výkon.

Obrázek 15: Průběhy na hotovém zdroji.



Obrázek 16: Fotka hlavní části zdroje.

Seznam obrázků

1	Blokové schéma spínaného zdroje [8]	3
2	Topologie DC/DC měničů a AC/DC zdrojů [1]	4
3	Propustné dvojčinné - Forward Push-Pull [2]	5
4	Feritová jádra	7
5	Vaření transformátoru	7
6	Převinutí transformátoru	7
7	Diody s chladičem	8
8	Arduino Mega 2560. Zdroj: http://arduino.cc/	9
9	Blokové schéma TL494. Zdroj: http://ti.com/	10
10	Blokové schéma	11
11	Schema	12
12	Deska plošných spojů	13
13	Konektor EC5	13
14	Vzhled GUI	14
15	Průběhy na hotovém zdroji.	21
16	Fotka hlavní části zdroje.	22

Slovník

AVR AVR je označení pro rodinu 8bitových a některých 32bitových mikročipů typu RISC s harvardskou architekturou od firmy Atmel.[3] 10

GPIO (General Purpose Input Output) je označení pinů, které lze použít jako vstupní i jako výstupní pro libovolnou aplikaci. 10

I²C (anglicky Inter-Integrated Circuit, čteme I-squared-C, nesprávně I-two-C) je multi-masterová počítačová sériová sběrnice vyvinutá firmou Philips, která je používána k připojování nízkorychlostních periférií k základní desce, vestavěnému systému nebo mobilnímu telefonu.[4] 10

MIPS Million Instruction Per Second (česky milion instrukcí za sekundu) je jednotka výkonnosti počítače, která udává počet zpracovaných instrukcí za sekundu. Alternativním označením je MOPS čili Million Operations Per Second (česky milion operací za sekundu).[5] 10

MOSFET Tranzistor MOSFET (Metal Oxide Semiconductor Field Effect Transistor) je polem řízený tranzistor, kde je vodivost kanálu mezi elektrodami Source a Drain ovládána elektrickým polem vytvářeným ve struktuře kov(M)-oxid(O)-polovodič(S) napětím přiloženým mezi hradlo (Gate) a Source. Hradlo je odděleno od polovodiče vrstvou oxidu křemíku - odtud oxid v názvu tohoto typu tranzistoru.[6] 7

SPI (Serial Peripheral Interface) je sériové periferní rozhraní. Používá se pro komunikaci mezi řídicími mikroprocesory a ostatními integrovanými obvody (EEPROM, A/D převodníky, displeje...).[7] 10

Reference

- [1] Jan Babčaník. Spínané zdroje. <http://www.hw.cz/teorie-a-praxe/spinane-zdroje.html>, 2013.
- [2] danyk.cz. Impulzní zdroje a měniče i. <http://danyk.cz/zdroje.html>, 2013.
- [3] Wikipedie. Avr — wikipedie: Otevřená encyklopedie. <http://cs.wikipedia.org/w/index.php?title=AVR&oldid=9922184>, 2013. [Online; navštíveno 31. 03. 2013].
- [4] Wikipedie. I²c — wikipedie: Otevřená encyklopedie. <http://cs.wikipedia.org/w/index.php?title=I%C2%B2C&oldid=9902359>, 2013. [Online; navštíveno 31. 03. 2013].
- [5] Wikipedie. Mips — wikipedie: Otevřená encyklopedie. <http://cs.wikipedia.org/w/index.php?title=MIPS&oldid=9932365>, 2013. [Online; navštíveno 31. 03. 2013].
- [6] Wikipedie. Mosfet — wikipedie: Otevřená encyklopedie. <http://cs.wikipedia.org/w/index.php?title=MOSFET&oldid=9866427>, 2013. [Online; navštíveno 1. 04. 2013].
- [7] Wikipedie. Serial peripheral interface — wikipedie: Otevřená encyklopedie. http://cs.wikipedia.org/w/index.php?title=Serial_Peripheral_Interface&oldid=9899620, 2013. [Online; navštíveno 31. 03. 2013].
- [8] Wikipedie. Spínaný zdroj — wikipedie: Otevřená encyklopedie. http://cs.wikipedia.org/w/index.php?title=Sp%C3%ADnan%C3%BD_zdroj&oldid=9915281, 2013. [Online; navštíveno 24. 03. 2013].