



**Střední průmyslová škola a obchodní akademie Uherský Brod**

Nivnická 1781, 68801 Uherský Brod

## **Závěrečná práce**

### **Nízkofrekvenční osciloskop**

Konzultant: **Ing. Josef Nevařil**

Autor: **Tomáš Janečka**

**V Uherském Brodě, 2016**

## **PROHLÁŠENÍ**

Prohlašuji, že jsem závěrečnou práci na téma nízkofrekvenční osciloskop vypracoval samostatně s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použité literatury a pramenů.

Uherský Brod 11.1.2017

Podpis

.....

Tomáš Janečka

## **Poděkování**

Děkuji mému konzultantovi panu Nevařilovi za poskytnutí rad ohledně softwaru i za nápady na vylepšení a že si na mě vždy udělal čas. Rád bych rovněž poděkoval panu mistrovi Kolářovi za poskytnutí prostředků k produkci desky plošných spojů.

**Střední průmyslová škola a obchodní akademie Uherský Brod**

Nivnická 1781, 68801 Uherský Brod

**Zadání:** Zkonstruovat generátor obdélníkových impulsů pomocí obvodu 555 a napsat program pro zobrazení signálu přivedeného na analogový port jednočipového počítače a snažit se, aby byl frekvenční rozsah co největší.

Tomáš Janečka  
Mechanik elektrotechnik  
10.5.2016  
Ing. Josef Nevařil

ME4 2016/2017  
Počítačová technika  
12.1.2017

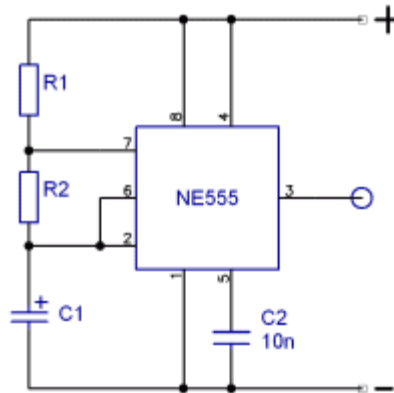
## **OBSAH**

|  |          |
|--|----------|
| <b>1.Úvod.....</b>                       | <b>6</b> |
| <b>2. Metodika práce .....</b>           | <b>7</b> |
| <b>2.1 Astabilní multivibrátor .....</b> | <b>7</b> |
| <b>2.1.1 Popis funkce .....</b>          | <b>8</b> |
| <b>2.1.2 DPS .....</b>                   | <b>9</b> |
| <b>2.2 Program – Arduino.....</b>        | <b>9</b> |

|   |           |
|---|-----------|
| <b>2.3 Program – Processing .....</b>               | <b>10</b> |
| <b>3. Výsledky měření .....</b>                     | <b>13</b> |
| <b>3.1 Multivibrátor .....</b>                      | <b>13</b> |
| <b>3.2 Osciloskop .....</b>                         | <b>13</b> |
| <b>4. Závěr .....</b>                               | <b>14</b> |
| <b>5. Seznam použité literatury a pramenů .....</b> | <b>15</b> |

# 1. ÚVOD

Má závěrečná práce se skládá ze dvou částí. První část je výroba astabilního multivibrátoru pomocí časovacího obvodu 555. Použiju k tomu klasické zapojení:



Rezistor R2 jsem nahradil potenciometrem, abych mohl měnit střihu a to aby se pak projevilo v grafu. Obvod 555 je napájen 5V kvůli jednočipovému počítači.

Druhá část se skládá z napsání programu pro zobrazení signálu v softwaru Processing. Zobrazované hodnoty získám na analogovém portu jednočipového počítače a následně je pošlu z Arduina IDE do Processingu přes sériový port.

Zadání spojuje několik věcí. Výroba hardware – multivibrátor, práce s jednočipovým počítačem a jeho naprogramování a taky vytvoření grafického prostředí pro osciloskop.

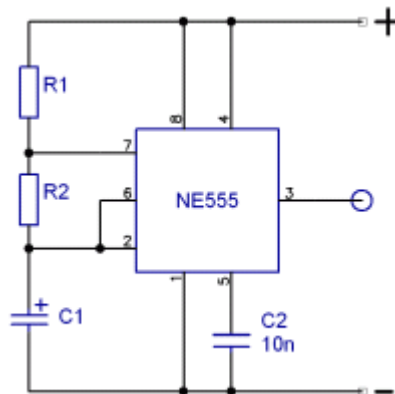
## 2. METODIKA PRÁCE

### 2.1 Astabilní multivibrátor

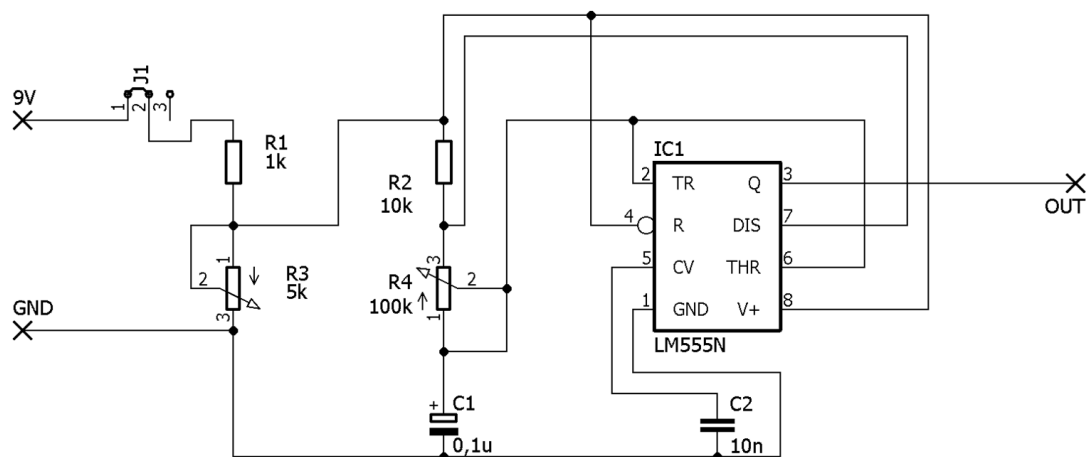
Jedná se o klopný obvod, který nemá ani jeden stabilní stav. Neustále se překlápí z jednoho do druhého stavu. Díky tomu na svém výstupu vytváří obdélníkový průběh.



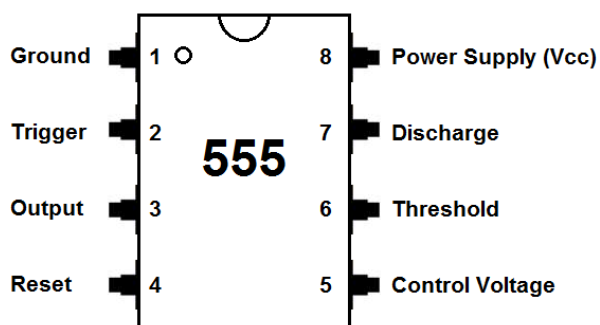
Tento obvod lze poskládat buď z tranzistorů nebo z časovacího obvodu 555, pro který jsem se rozhodl. Zde je schéma, ze kterého jsem vycházel:



Časovací obvod 555 v tomto konkrétním zapojení funguje na principu počítání, za jak dlouho se nabije a vybije elektrolytický kondenzátor a na základě této doby generuje na výstupu napětí. Hodnotami C1, R1 a R2 můžeme měnit tzv. střidu. Střida je poměr časů, ve kterých je obdélníkový signál v jednotlivých úrovních. Pevnými hodnotami součástek dosáhneme pořád stejné střidy. Pro její změnu jsem se rozhodl místo R2 umístit potenciometr. Jelikož pracuji s jednočipovým počítačem, jehož vstup je schopen zpracovávat napětí do 5V SS, je potřeba, aby byl časovací obvod napájen 5V. Vyřešil jsem to děličem napětí napájeného 9V baterií.



### 2.1.1 Popis funkce



1 – Ground – nula

2 – Trigger – připojením tohoto pinu do nuly začne časování

3 – Output – výstup

4 – Reset – připojením pinu do nuly se časování resetuje

5 – CV – k pinu je většinou připojen kondenzátor o malé kapacitě, slouží ke stabilizaci napájecího napětí aby časování probíhalo bez chyb

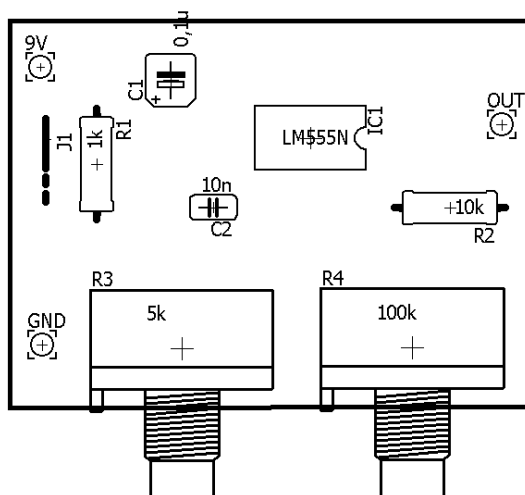
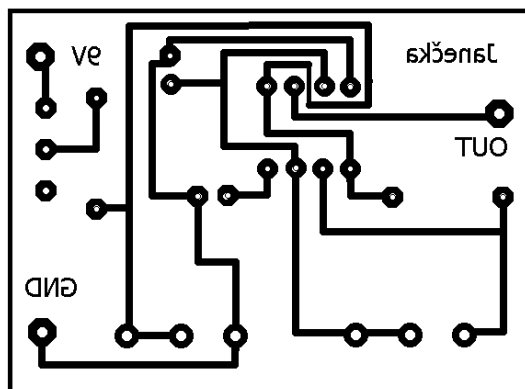
6 – Threshold – pin sleduje napětí na kondenzátoru a počítá čas nabití do asi 2/3 kapacity

7 – Discharge – vybíjí kondenzátor když je plný

Pro funkci obvodu 555 v astabilním módu jsou spojeny piny 2 a 6. Po připojení napájení se kondenzátor nabíjí přes dva rezistory a vybíjí se pouze přes jeden. To znamená, že signál na výstupu bude o něco málo déle v logické úrovni 1. Jakmile pin 6 detekuje nabitý kondenzátor, nastává jeho vybíjení přes pin 7. Když je kondenzátor vybitý, pin 2 detekuje nulu a kondenzátor se opět začne nabíjet.

### 2.1.2 DPS





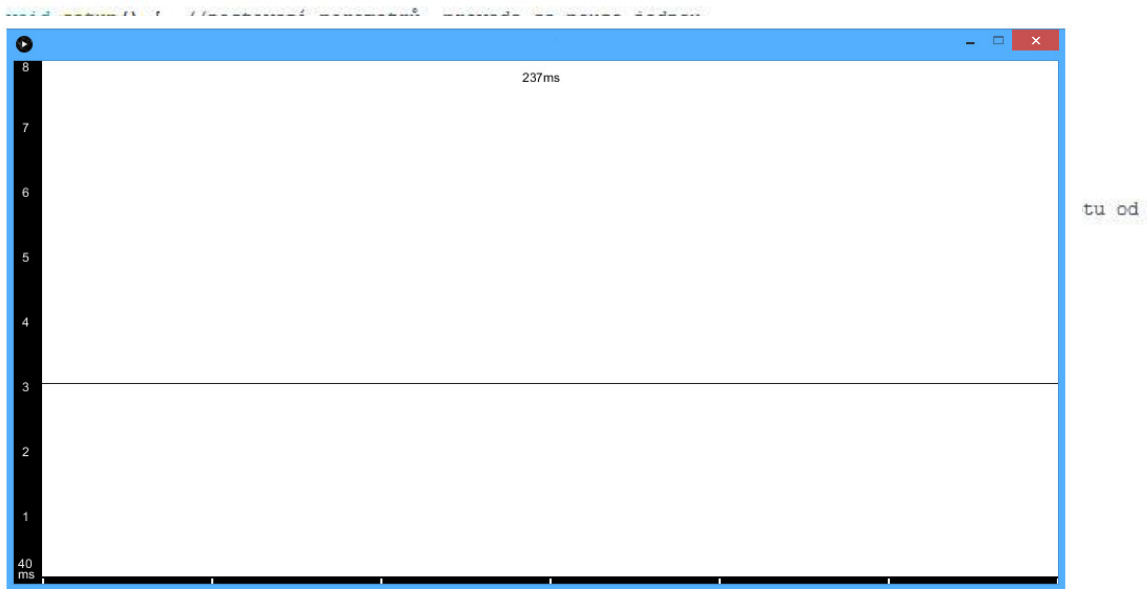
Potenciometr vlevo je použit k nastavení napájecího napětí. Otáčením doprava se napětí snižuje.

Potenciometr vpravo je určen k nastavení kmitočtu. Otáčením doprava se kmitočet zvyšuje.

## 2.2 Program – Arduino

Využívám jednočipový počítač Arduino Nano jako převodník napětí na číslo. Výstup z nestabilního multivibrátoru je připojen na analogový pin Arduina. Hodnoty napětí posílám přes sériovou komunikaci do programu Processing, kde jsou následně data zpracovány a zobrazeny. Rychlost přenosu je nastavena na 250 000 baudů.

## 2.3 Program – Processing



Prostředí osciloskopu je můj vlastní přínos. Je navrženo tak, že hodnota napětí přivedeného signálu může být změřena na stupnici u levého okraje.

```
import processing.serial.*; //import knihovny pro sériovou komunikaci
Serial port; //vytvoření class Serial a pojmenování port

float hodnota = 0.0; //deklarování proměnné pro zápis naměřených hodnot
PImage obrazek; //deklarování proměnné pro obrázek

void setup()
{
  size(1000, 500); //vytvoření okna o výšce 500 px a šířce 1000 px
  obrazek = loadImage("NFosc.jpg"); //nahrání obrázku do proměnné

  port = new Serial(this, "COM4", 250000); //zahájení sériové komunikace, výběr portu ze kterého má číst
  //a nastavení rychlosti komunikace
  port.bufferUntil('\n'); //číst ze sériového portu dokud se neobjeví znak \n = odřádkování

  image(obrazek,0,0); //zobrazit obrázek na souřadnice 0,0
}
```

Na začátku programu importuju knihovny a vytvářím class Serial. V části setup nastavuji velikost okna, sériovou komunikaci a vykresluju obrázek. Místo položky „COM4“ je potřeba nastavit správný port. Ten se zjistí při připojení Arduina k PC.

```

void draw(){ //obdoba smyčky loop v Arduino IDE

  fill(0); //nastavení černé barvy
  if(millis(>2000){ //podmínka pokud je doba od začátku programu větší než 2 sekundy
    for(int i=26;i<1000;i++){ //smyčka pro vykreslování naměřených hodnot
      point(i,hodnota); //bod o souřadnicích x a y
      if(i==26){ //pokud se pomocná proměnná rovná 26
        fill(255); //nastavení bílé barvy
        rect(26,-1,974,494); //vykreslení obdélníku na souřadnicích x = 26 a y = -1
                                //o šířce 974 px a výšce 494 px kvůli zobrazení nově příchozích
                                //naměřených hodnot
        fill(0); //nastavení černé barvy
        text("237ms",487,20); //vypsání textu 237 ms na souřadnice x = 487 a y = 20
      }
    }
  }
  else { //pokud podmínka není splněna
    delay(1000); //zastavit chod programu na 1 sekundu
    pozadi(); //volání funkce pozadí()
  }
}

```

Část pro zobrazení naměřených hodnot do grafu.

```

void pozadi(){ //definice funkce pro vykreslení pozadí
  background(255); //bílá barva pozadí
  fill(0); //nastavení černé barvy
  rect(0,0,26,500); //vykreslení obdélníku na souřadnicích x = 0 a y = 0 o šířce 26 px a výšce 500 px
  rect(20,494,1000,6); //vykreslení obdélníku na souřadnicích x = 20 a y = 494 o šířce 1000 px a výšce 6 px
  text("237ms",487,20); //vypsání textu 237 ms na souřadnice x = 487 a y = 20
  fill(255); //nastavení bílé barvy
  text("1",8,440); //vypsání textu 1 na souřadnice x = 8 a y = 440
  text("2",8,378); //vypsání textu 2 na souřadnice x = 8 a y = 378
  text("3",8,316); //vypsání textu 3 na souřadnice x = 8 a y = 316
  text("4",8,254); //vypsání textu 4 na souřadnice x = 8 a y = 254
  text("5",8,192); //vypsání textu 5 na souřadnice x = 8 a y = 192
  text("6",8,130); //vypsání textu 6 na souřadnice x = 8 a y = 130
  text("7",8,68); //vypsání textu 7 na souřadnice x = 8 a y = 68
  text("8",8,10); //vypsání textu 8 na souřadnice x = 8 a y = 10
  text("40",4,485); //vypsání textu 40 na souřadnice x = 4 a y = 485
  text("ms",4,495); //vypsání textu ms na souřadnice x = 4 a y = 495
  for(int i=26;i<1000;i=i+162){ //smyčka pro vykreslení čar na ose x
    rect(i,494,3,6); //vykreslení obdélníku na souřadnicích x = i a y = 494 o šířce 3 px a výšce 6 px
  }
}

```

Funkce pro zobrazení pozadí je použita v části draw pro nastavení prostředí osciloskopu.

```

void serialEvent (Serial port){ //funkce se zavolá když se na sériovém portu objeví data

    String y = port.readStringUntil('\n'); //deklarace stringu y, do kterého se zapisují
                                        //data ze sériového portu dokud nenarazí na znak \n

    if (y != null) { //pokud je v proměnné y platný datový typ
        y = trim(y); //odstranit přebytečné bílé místo před nebo za daty přijatými do proměnné
        hodnota = float(y); //zapsání dat do proměnné hodnota, které se zapisou jako float
        hodnota = map(hodnota, 0.0, 1023.0, height - 8.0, height - 310.0); //přenastavení rozsahu přijatých dat
        //přenastavení proměnné hodnota, přijatá data ve tvaru od 0.0 do 1023.0 převést na tvar od height(šířka) - 8.0 do height - 310.0
    }
}
}

```

Část pro příjem hodnot poslaných po sériové lince a jejich úprava.

## 3. VÝSLEDKY MĚŘENÍ

### 3.1 Multivibrátor

Z astabilního klopného obvodu v konečné podobě vystupuje napětí o obdélníkovém průběhu s frekvenčním rozsahem od asi 50 Hz do 2 kHz a výstupní napětí je nastavitelné od asi 2V do 8V. Frekvenční rozsah byl ověřen na analogovém osciloskopu. Při nastavení nejnižšího kmitočtu je střída asi 50%, ovšem při nejvyšším nastaveném kmitočtu je asi 10%.

### 3.2 Osciloskop

Naprogramovaný osciloskop je schopen zobrazovat signál s hodnotami od 0V do 5V SS a od 0 Hz do asi 2,5 kHz. Vypsáním funkce millis() na sériový port v programu Arduino IDE jsem zjistil, že za 1 ms se pošlou dvě hodnoty, z čehož vyplývá, že jedna se pošle za 0,5 ms. V programu Processing na každý pixel vypisují jednu přijatou hodnotu, takže na osciloskopu vidíme časový průběh signálu za 237 ms.

## 4. ZÁVĚR

Jako závěrečnou práci se mi podařilo vyrobit generátor obdélníkového signálu a napsat program pro zobrazení hodnot přivedených na sériový port. Pár problémů se ovšem vyskytlo. Při návrhu DPS pro multivibrátor jsem počítal s tím, že použiju jednoduchý vypínač. Umístil jsem ovšem na desku špatné plošky a vypínač byl pro toto rozpětí příliš malý, tak jsem byl nucen na místo vypínače umístit drátový propoj. Multivibrátor se nyní zapíná a vypíná připojením nebo odpojením baterie. Dále jsem na DPS zapomněl umístit plošku pro uzemění výstupu a proto jsem musel na DPS zespodu připájet vývodní kabel. Softwarově osciloskop postačuje orientačním měřením pro nenáročného uživatele. Jako zlepšení a zvětšení možností osciloskopu je možné k jednočipovému počítači připojit předřadný rezistor a zvětšit jím napět'ový rozsah. Dále je možné se úplně vyvarovat volání funkce `analogRead()` a programovat jednočipový počítač na úrovni Assembleru. Při použití časování, interruptů, `free-running` modu A/D převodníku je možné mnohonásobně zvětšit rychlost vzorkování.

## **5. Seznam použité literatury a pramenů**

[http://pandatron.cz/?490&555\\_a\\_556\\_-\\_zakladni\\_zapojeni](http://pandatron.cz/?490&555_a_556_-_zakladni_zapojeni)

<https://www.arduino.cc/en/Tutorial/HomePage>

<https://www.arduino.cc/en/Reference/HomePage>

<https://processing.org/tutorials/>

<https://processing.org/reference/>