

## REALIZACE ELEKTRONICKÝCH ZAŘÍZENÍ NA BÁZI PLATFORMY ARDUINO

### IMPLEMENTATION OF ELECTRONIC DEVICES BASED ON ARDUINO PLATFORM

PAVEL ESCHLER

#### **Resumé**

*V praktické části práce jsem se snažil podat modifikované existující projekty co nejjednodušší formu pro čtenáře. Praktická část lze považovat za rychlý náhled do tajů programování desek řízených mikrokontrolérem. Jednotlivé projekty jsou řazeny vzestupně dle obtížnosti. Poslední projekt Dotykové piano jsem realizoval s ohledem na možné využití ve vzdělávání. Hudba a technika nejsou příliš spojovanými směry, i přesto technika může mít přímou spojitost s výukou hudby a naopak.*

#### **Abstract**

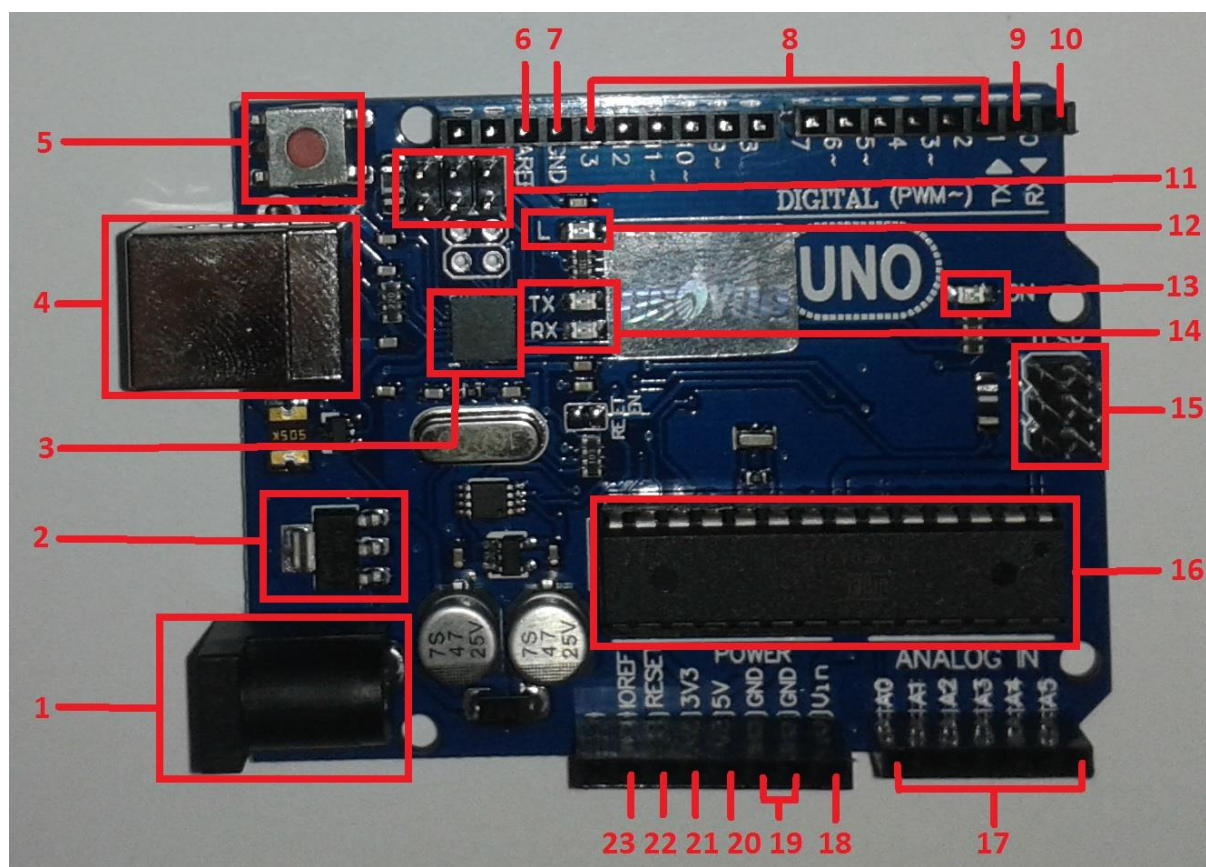
*In practical part I tried to demonstrate modified, already existing schemes as easy as it possible. Practical part can be taken as the quick preview of programming boards controlled by microcontroller. All the projects are sorted gradually from easiest to hardest. I designed the last scheme Touch piano, for possible use in education. Music and technology might not seem much connected, despite technology can be used in music teaching.*

#### **ÚVOD**

Cílem práce je shrnout poznatky dosažené v jednotlivých kapitolách a vytvořit díky nim komplexní úlohu simulující hudební nástroj - dotykové piano. Práce má zároveň za cíl popularizovat ve školství využití programování za pomoci fyzického výstupu, nejen u technických oborů ale i netechnických jako je například hudební výchova.

#### **REALIZACE DOTYKOVÉHO PIANA**

Projekt dotykového piana byl realizován na klonu desky Arduino Uno R3. Osoyoo Uno R3 je sestaveno dle open-source hardwarového designu, za použití stejných součástek. Deska je díky tomu věrnou kopií originálního sestavení desky Arduino Uno R3, od které se funkčně nijak neliší. Drobné rozdíly jsou pouze v barvě samotné desky a umístění některých popisů na desce.



Obrázek 21: Osoyoo UNO R3

1 - Konektor pro externí napájení	13 - Power LED (ON)
2 - Napěťový regulátor	14 - RX a TX LED
3 - Mikrokontrolér ATmega16U2	15 - ICSP ATmega328
4 - USB konektor typu B	16 - Mikrokontrolér ATmega328
5 - Tlačítko RESET	17 - Vstupní analogové piny
6 - AREF pin	18 - Vstupní napájecí pin
7 - GND pin	19 - GND piny
8 - Vstupní/výstupní digitální piny	20 - Výstupní napájecí 5V pin
9 - Pin 1(TX)	21 - Výstupní napájecí 3.3V pin
10 - Pin 0(RX)	22 - RESET pin
11 - ICSP ATmega16U2	23 - IOREF pin
12 - LED	

Tabulka 1: Vysvětlivky k číselnému značení

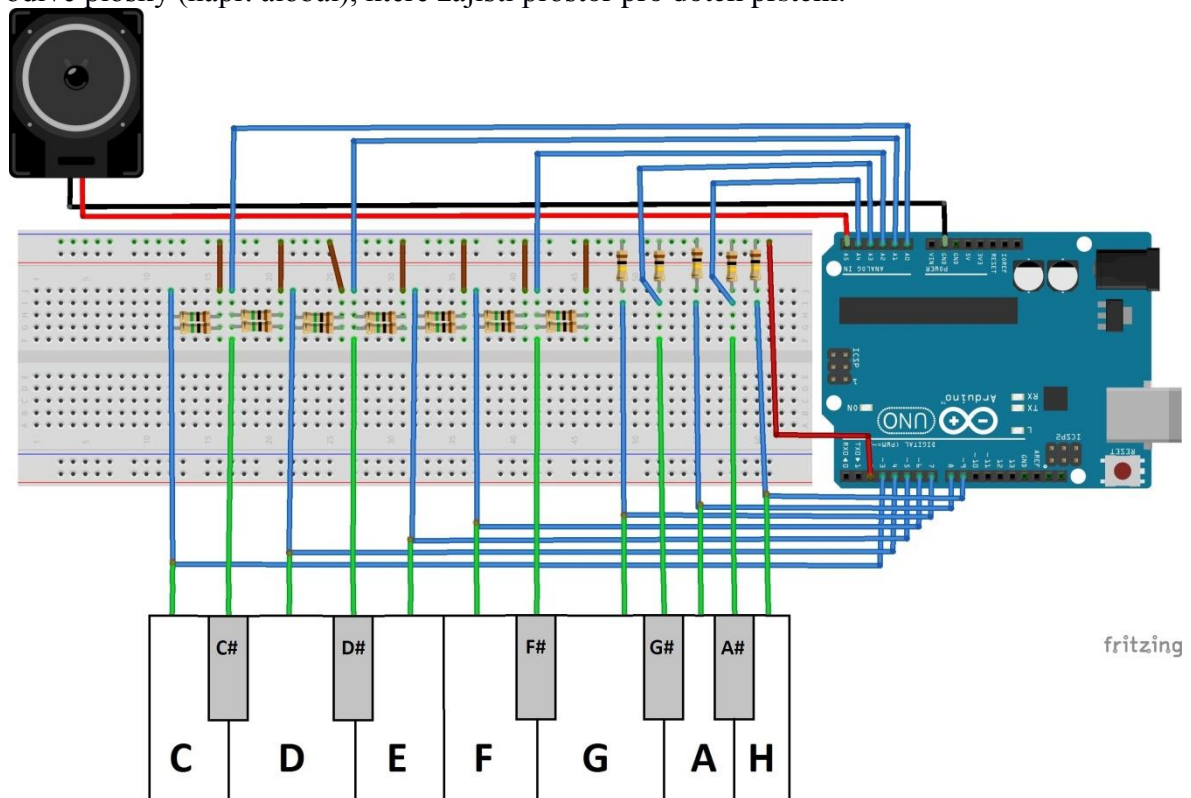
## POPIS PROJEKTU

Projekt simuluje funkci piana pomocí dotykových ploch, vytvořených z alobalu nebo jiného vodivého materiálu. Ke snímání dotyku, bez nutnosti dotykového kapacitního modulu, využívá knihovnu CapacitiveSensor, která nastavuje piny jako senzory s režimem kapacitní citlivosti. Instalace jednotlivých senzorů vyžaduje odpory s rezistencí od 50k ohm do 50M ohm. Při zvyšující hodnotě rezistence, je senzor citlivější na přítomnost lidského těla. Senzory jsou schopné díky různým rezistencím rozeznat přímý kontakt s dotykovou plochou (1Mohm) a nebo pouhé přiblížení k dotykové ploše (10Mohm a více dle požadované vzdálenosti). Hlavní předností tohoto projektu je využití všech analogových vstupů v digitálním režimu.

Díky těmto pinům je možné simulovat celý rozsah oktávy (7 celých not + 5 půl not). Přečtená hodnota senzoru se programově porovnává s mezí, pokud je mez překročena, program vyhodnotí vstup jako dotknutí se. Dle senzoru, jehož hodnota překročila mez, přehraje po stanovenou dobu při stanovené frekvenci odpovídající tón pomocí připojeného reproduktoru. V zapojení jsou použity dva druhy rezistorů, základní má rezistenci 1Mohm, zbývající 2Mohm - tyto rezistory byly použity z důvodu nedostatku 1Mohm rezistorů, jsou zapojeny paralelně, výsledná rezistence je tedy dle vzorce  $R1=(R2+R2)/(R2*R2)=4/4=1[\text{Mohm}]$ . Podrobné informace k fungování knihovny CapacitiveSensor a významu některých funkcí jako například millis() je možné dohledat na oficiálních stránkách Arduina (<http://playground.arduino.cc//Main/CapacitiveSensor?from=Main.CapSense>). Velikosti rezistorů důrazně nedoporučuji kombinovat, jelikož se vstupy kalibrují dle hodnot mezi piny 2-3, je nutné stejné hodnoty dodržet v celém zapojení.

### POPIS ZAPOJENÍ

Na desku nepájivého kontaktního pole připojíme pin 2 na svislíci užívanou pro společnou zem, k této svislíci postupně připojíme rezistory 1Mohm, v případě 2Mohm rezistorů tyto rezistory po dvou připojíme paralelně (vznikne tak rezistence mezi konci o požadované velikosti 1Mohm). Druhý vývod rezistorů postupně připojíme k digitálním pinům 3 až 9 (celé noty) a analogovým vstupům nastavených v digitálním režimu A0 až A4, neboli 14 až 19 (půlené noty). Mezi vývody rezistorů a piny připojíme pomocí vodičů jakékoliv vodivé plošky (např. alobal), které zajistí prostor pro dotek prstem.



Obrázek 22: Zapojení dotykového pianina

## KÓD PROGRAMOVÉHO ŘÍZENÍ

```
/* Autor Pavel Eschler
   Arduino Uno piano s kompletním rozsahem tónu 5té oktávy za pomoci analogových vstupů A0 až
   A5 = digitální PWM vstupy 14 až 19.
   Celý projekt zabírá 7 036 bytů úložného místa pro program.
   Globální proměnné zabírají 583 bytů dynamické paměti.
*/
#include <CapacitiveSensor.h> //Importování knihovny CapacitiveSensor Library.

#define reproduktor 19 //definuje proměnnou reproduktor na pinu A5=19 jako LED, tedy
výstupní, lze použít jeden z neobsazených PWM pinu, například 11

// definuje pin 2 pro odesílání a pin 3 až 18 pro příjem
// piny 2-9 jsou použity jako základní tóny
CapacitiveSensor cs_2_3 = CapacitiveSensor(2, 3); // mezi piny 2 a 3 je 1Mohm rezistor, pin
3 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_4 = CapacitiveSensor(2, 4); // mezi piny 2 a 4 je 1Mohm rezistor, pin
4 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_5 = CapacitiveSensor(2, 5); // mezi piny 2 a 5 je 1Mohm rezistor, pin
5 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_6 = CapacitiveSensor(2, 6); // mezi piny 2 a 6 je 1Mohm rezistor, pin
6 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_7 = CapacitiveSensor(2, 7); // mezi piny 2 a 7 je 1Mohm rezistor, pin
7 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_8 = CapacitiveSensor(2, 8); // mezi piny 2 a 8 je 1Mohm rezistor, pin
8 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_9 = CapacitiveSensor(2, 9); // mezi piny 2 a 9 je 1Mohm rezistor, pin
9 funguje jako dotykový senzor při připojení kovového kontaktu

// piny 14-18 jsou použity jako pultóny
CapacitiveSensor cs_2_14 = CapacitiveSensor(2, 14); // mezi piny 2 a 14 je 1Mohm rezistor,
pin 14 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_15 = CapacitiveSensor(2, 15); // mezi piny 2 a 15 je 1Mohm rezistor,
pin 15 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_16 = CapacitiveSensor(2, 16); // mezi piny 2 a 16 je 1Mohm rezistor,
pin 16 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_17 = CapacitiveSensor(2, 17); // mezi piny 2 a 17 je 1Mohm rezistor,
pin 17 funguje jako dotykový senzor při připojení kovového kontaktu
CapacitiveSensor cs_2_18 = CapacitiveSensor(2, 18); // mezi piny 2 a 18 je 1Mohm rezistor,
pin 18 funguje jako dotykový senzor při připojení kovového kontaktu

void setup()
{
  cs_2_3.set_CS_Autocal_Millis(0xFFFFFFFF); // vypne autokalibraci na kanálu 1
  Serial.begin(9600); //inicializace sériové komunikace, parametr určuje počet přenášených
bitů za sekundu, v tomto případě 9600
}

void loop()
{
  long start = millis(); //nastaví časovač, millis je rychlejší alternativa k delay

  // přiřazení vstupů k proměnným
  long notaC = cs_2_3.capacitiveSensor(60); //načte hodnotu senzoru na pinu 3, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
  long notaD = cs_2_4.capacitiveSensor(60); //načte hodnotu senzoru na pinu 4, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
  long notaE = cs_2_5.capacitiveSensor(60); //načte hodnotu senzoru na pinu 5, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
  long notaF = cs_2_6.capacitiveSensor(60); //načte hodnotu senzoru na pinu 6, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
  long notaG = cs_2_7.capacitiveSensor(60); //načte hodnotu senzoru na pinu 7, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
  long notaA = cs_2_8.capacitiveSensor(60); //načte hodnotu senzoru na pinu 8, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
  long notaH = cs_2_9.capacitiveSensor(60); //načte hodnotu senzoru na pinu 9, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
```

```
long notaCs = cs_2_14.capacitiveSensor(60);//načte hodnotu senzoru na pinu 14, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
long notaDs = cs_2_15.capacitiveSensor(60);//načte hodnotu senzoru na pinu 15, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
long notaFs = cs_2_16.capacitiveSensor(60);//načte hodnotu senzoru na pinu 16, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
long notaGs = cs_2_17.capacitiveSensor(60);//načte hodnotu senzoru na pinu 17, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více
long notaAs = cs_2_18.capacitiveSensor(60);//načte hodnotu senzoru na pinu 18, parametr 60
označuje citlivost čtení a nastavuje se v závislosti na použitém rezistoru, pro 10Mohm je
doporučeno 80 a více

Serial.print(millis() - start); // zkontroluje výkon v milisekundách
// tisk hodnot na monitor
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaC); // vytiskne hodnotu senzoru připojeného k pinu 3
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaD); // vytiskne hodnotu senzoru připojeného k pinu 4
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaE); // vytiskne hodnotu senzoru připojeného k pinu 5
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaF); // vytiskne hodnotu senzoru připojeného k pinu 6
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaG); // vytiskne hodnotu senzoru připojeného k pinu 7
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaA); // vytiskne hodnotu senzoru připojeného k pinu 8
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaH); // vytiskne hodnotu senzoru připojeného k pinu 9
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor

Serial.print(notaCs); // vytiskne hodnotu senzoru připojeného k pinu 14
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaDs); // vytiskne hodnotu senzoru připojeného k pinu 15
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaFs); // vytiskne hodnotu senzoru připojeného k pinu 16
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.print(notaGs); // vytiskne hodnotu senzoru připojeného k pinu 17
Serial.print("\t"); // vytiskne na obrazovku sériového monitoru odsazení -
tabulátor
Serial.println(notaAs); // vytiskne hodnotu senzoru připojeného k pinu 18, ukončí
řádek, další výpis proběhne od začátku nového řádku

// pokud se dotkneme kovového kontaktu připojeného mezi čtecí pin a rezistor, hodnota
překročí mez a rozezná reproduktor, jedná se o další nastavení citlivosti
// frekvence 523Hz odpovídá notě C 5té oktávy, další noty jsou opět frekvenčně nastaveny dle
5-té oktávy, parametry frekvence a doba znění lze libovolně měnit
if (notaC > 150) tone(reproduktor, 523, 250); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 250ms bude při frekvenci 523Hz střídát napětí 0 a 5V
if (notaD > 150) tone(reproduktor, 587, 250); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 250ms bude při frekvenci 587Hz střídát napětí 0 a 5V
if (notaE > 150) tone(reproduktor, 659, 250); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 250ms bude při frekvenci 659Hz střídát napětí 0 a 5V
if (notaF > 150) tone(reproduktor, 698, 250); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 250ms bude při frekvenci 698Hz střídát napětí 0 a 5V
if (notaG > 150) tone(reproduktor, 784, 250); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 250ms bude při frekvenci 784Hz střídát napětí 0 a 5V
if (notaA > 150) tone(reproduktor, 880, 250); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 250ms bude při frekvenci 880Hz střídát napětí 0 a 5V
if (notaH > 150) tone(reproduktor, 988, 250); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 250ms bude při frekvenci 988Hz střídát napětí 0 a 5V
```

```
//frekvence 554Hz odpovídá frekvenčně C# 5té oktávy
if (notaCs > 150) tone(reproduktor, 554, 150); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 150ms bude při frekvenci 554Hz střídat napětí 0 a 5V
if (notaDs > 150) tone(reproduktor, 622, 150); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 150ms bude při frekvenci 622Hz střídat napětí 0 a 5V
if (notaFs > 150) tone(reproduktor, 740, 150); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 150ms bude při frekvenci 740Hz střídat napětí 0 a 5V
if (notaGs > 150) tone(reproduktor, 831, 150); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 150ms bude při frekvenci 831Hz střídat napětí 0 a 5V
if (notaAs > 150) tone(reproduktor, 932, 150); // pokud je překročena mez, aktivuje pin
reproduktoru a po dobu 150ms bude při frekvenci 932Hz střídat napětí 0 a 5V

// vypne genereování tónu do reproduktoru, lze jím nastavit délku generování tónu hromadně
pro všechny noty
/* začátek blokového komentáře
if (notaC<=150 & notaD<=150 & notaE<=150 & notaF<=150 & notaG<=150 & notaA<=150
& notaH<=150 & notaCs<=150 & notaDs<=150 & notaFs<=150 & notaGs<=150 & notaAs<=150)
{
    delay(250);
    noTone(reproduktor);
}
konec blokového komentáře */
delay(10); // pauza definující rychlost čtení, omezuje tok dat na monitor sériového portu
a zajišťuje stabilitu čtení
}
```

## ZÁVĚR

Platforma Arduino umožňuje za velmi přívětivou cenu realizovat širokou škálu projektů, díky kterým je uživatel o krok blíže k pochopení fungování programově řízených obvodů a počítačů. Za hlavní přínos práce považuji ověření Osoyoo klonu desky Arduino, která dokázala, že není nutné kupovat vždy nejdražší součástky a příslušenství. Osoyoo Uno klon dosahuje v projektech stejných výsledků jako originální deska, přičemž tento nebo jiný klon se dá pořídit i za desetinu ceny originální desky.

Arduino si díky své jednoduchosti a fyzickému výstupu postupně razí cestu ve vzdělávacích programech po celém světě. Platforma především vzbuzuje zájem o techniku a poukazuje na fakt, že kód který uživatel naprogramuje nemusí nutně zůstat uvnitř osobního počítače a být tak pro většinu uživatelů jen nutnou překážkou. Díky vizualizaci kódu prostřednictvím vývojové desky, dochází u uživatele ke zvýšení zájmu o programování jako takové.

## LITERATURA

1. *Arduino* [online]. 2016 [cit. 2016-02-29]. Dostupné z: <http://www.arduino.cc/>
2. NUSSEY, John. *Arduino for dummies*. West Sussex, England: Wiley, c2013. --For dummies. ISBN 978-1-118-44642-3.
3. VODA, Zbyšek. *Průvodce světem Arduina*. Vydání první. Bučovice: Martin Stríž, 2015. ISBN 978-80-87106-90-7.
4. MARGOLIS, Michael. *Arduino cookbook*. 2nd ed. Sebastopol, Calif.: O'Reilly, c2012. ISBN 14-493-1387-6.
5. UDANIS, ALEX. *5 Great Arduino Alternatives*. In: All about circuits [online]. Boise: EETech Media, 2015 [cit. 2016-02-29]. Dostupné z: <http://www.allaboutcircuits.com/news/5-great-arduino-alternatives/>
6. BANZI, Massimo a Michael SHILOH. *Getting Started with Arduino*. USA. Maker Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2014. ISBN 978-1-4493-6333-8.

*Olympiáda techniky Plzeň 2016, 17. 5. 2016*  
*www.olympiadatechniky.zcu.cz*

7. MONK, Simon. *30 Arduino projects for the evil genius*. New York: McGraw-Hill, c2010, xiii, 191 p. ISBN 00-717-4133-X.

### **Kontaktní adresa**

**Pavel Eschler**, Katedra výpočetní a didaktické techniky

Fakulta pedagogická Západočeské univerzity v Plzni

E-mail: [eschler.pavel@seznam.cz](mailto:eschler.pavel@seznam.cz)