

1. LEKCE: PRVNÍ PROGRAM

LEKCE JE ZAMĚŘENA NA PRVNÍ SEZNÁMENÍ S PROGRAMOVATELNOU DESKOU ARDUINO, KTERÁ JE SOUČÁSTÍ DOPORUČENÉ SADY. VYUŽÍVÁ SE K TOMU JEDNODUCHÝCH OBVODŮ S LED DIODOU A BZUČÁKEM.

CÍLE

- ① Co je digitální vstup a výstup.
- ② Ovládání LED diody pomocí desky Arduino.
- ③ První seznámení s programovacím jazykem Wiring a jeho základní strukturou.
- ④ Seznámení s programem Arduino IDE pro nahrávání kódu do desky.
- ⑤ Princip a zapojení piezzo bzučáku.
- ⑥ Zvládnutí programování ovládání LED diody a bzučáku.

Čas: **2x45 min**

Úroveň: ■■■■■

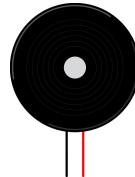
Vychází z: **1**



LED



Rezistor 220Ω



Piezzo bzučák

POUŽITÉ SOUČÁSTKY

PRŮVODCE HODINOU I



Studenti sestaví základní obvod, ve kterém použijí LED diodu s rezistorem. Tento obvod naprogramují podle vzorového příkladu a samostatných úkolů. Měli by se především seznámit s deskou Arduino a kontaktním polem. Naučí se postup při psaní programu a jeho nahrávání do desky.



PŘÍPRAVA

Co bude v této hodině potřeba?

- ① Součásti obvodu – deska Arduino s USB kabelem, kontaktní pole, 1x LED dioda, 1x rezistor 220Ω, 2x vodiče typu samec-samec.
- ② Osobní počítač pro studenty s nainstalovaným Arduino IDE.
- ③ Pokud je k dispozici, tak dataprojektor.
- ④ Prezentace k lekci 1.
- ⑤ Pracovní listy pro studenty.

1. KROK ⏰ 10 minut

Na úvod rozdejte studentům sady Arduino. Řekněte, že náplní vašeho kurzu bude naučit se základům programování ovládání elektronických obvodů, které řídí mikrokontrolér. Ty pak mohou tvorit části robotických systémů, inteligentních domů, ovládání automobilů apod.

ZEPTEJTE SE STUDENTŮ

➔ Víte, co to jsou embedded systémy?

Embedded systémy (ES) – někdy se uvádí vestavěné systémy, jsou kombinací hardwarového a softwarového vybavení. Jsou to systémy jednoúčelové, určené pro definované činnosti. ES bývají součástí nějakého systému většího. Obsahují řídící počítač, který je zcela zabudován do zařízení, které ovládá a plní konkrétní účel.



➔ Kde všude se s embedded systémy můžete potkat?

Dnes již prakticky všude: televize, mobilní telefony, automobily atd.

Studenti si sady mohou prohlédnout. Vysvětlete, že se s jednotlivými částmi budou postupně seznamovat na praktických příkladech. Není nutné všechny součásti sady Arduino vysvětlovat.

Ukažte jim pouze desku Arduino a řekněte, že tato deska obsahuje vstupy a výstupy, kterým se říká **piny** a srdcem desky je procesor (mikrokontrolér).

2. KROK ⏱ 10 minut

Nyní přistupte k sestavení prvního obvodu. Schéma obvodu můžete promítat pomocí dataprojektoru, nebo jej studenti sestaví podle pracovního listu. Vysvětlete jim princip kontaktního pole. Upozorněte studenty na to, že dioda má kratší a delší vývod. Na delší vývod musí být přivedeno napětí a na kratší zem z desky Arduino. Napětí se přivádí z výstupního pinu, který budeme ovládat programem.

Řekněte studentům, že LED dioda se může poškodit, proto se k napájení přidává rezistor.

3. KROK ⏱ 10 minut

Ať si studenti spustí Arduino IDE a napíší základní program.



Uvedený příklad studenti ani nemusí psát, stačí využít hotového příkladu. V Arduino IDE stačí otevřít **File > Examples > Basic > Blink**. Otevře se základní programový kód.

Řekněte studentům, aby připojili USB kabel k desce a do počítače. Pokud mají program připravený, ukažte jim, jak mají nastavit v rozhraní Arduino IDE desku, do které budou kód nahrávat, a port.

Kliknutím na ikonu pro upload kódu ať nahrají program do desky Arduino.



ZEPTEJTE SE STUDENTŮ

➔ **Co se děje s diodou, pokud je program v pořádku nahrán?**

Dioda bliká v intervalu jedné sekundy.

Studentům vysvětlete programový kód, zejména pak základní strukturu programu a použité funkce pro zápis hodnot na pinu desky.

4. KROK 15 minut

Pokud studenti vše zvládli, mohou řešit **samostatně úkoly**. Pokud je nestihnou v této hodině, mohou v nich pokračovat v hodině další.

ÚKOLY PRO STUDENTY

- ➔ Připojte druhou LED diodu, třeba na pin 12.
- ➔ Sestavte program pro střídavé blikání obou LED diod, vždy po 1 sekundě.
- ➔ Sestavte program pro střídavé blikání LED diod, ale tentokrát každá blikne dvakrát, vždy po 1 sekundě. Mezi oběma diodami bude pauza dvě sekundy.
- ➔ Upravte program tak, aby diody blikaly postupně, tj. rozsvítí se první, po 0,25s se rozsvítí druhá, v čase 0,5s zhasne první a v čase 0,75 shasne druhá. V čase 1 sekundy se rozsvítí první a pořád dokola.



PRACOVNÍ LIST – LED DIODA

PRVNÍ SEZNÁMENÍ SE SESTAVOVÁNÍM ELEKTRONICKÝCH OBVODŮ A PROGRAMOVÁNÍM DESKY ARDUINO. V TÉTO ČÁSTI VYTVOŘÍTE JEDNODUCHÝ ELEKTRONICKÝ OBVOD S LED DIODOU.

CO SE NAUČÍTE

- ① Základní použití desky Arduino.
- ② Princip a používání kontaktního pole.
- ③ Zapojení jednoduchého obvodu.
- ④ Vytvoření prvního programu pro ovládání LED diody.



CO BUDETE POTŘEBOVAT

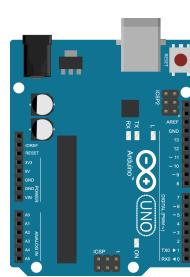
- ① LED diodu.
- ② Rezistor 220Ω .
- ③ Desku Arduino.
- ④ Kontaktní pole.
- ⑤ Vodiče typu samec-samec.



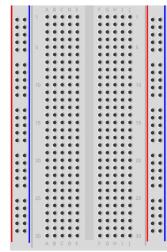
Led



Rezistor 220Ω



Deska Arduino



Kontaktní pole

POUŽITÉ SOUČÁSTKY

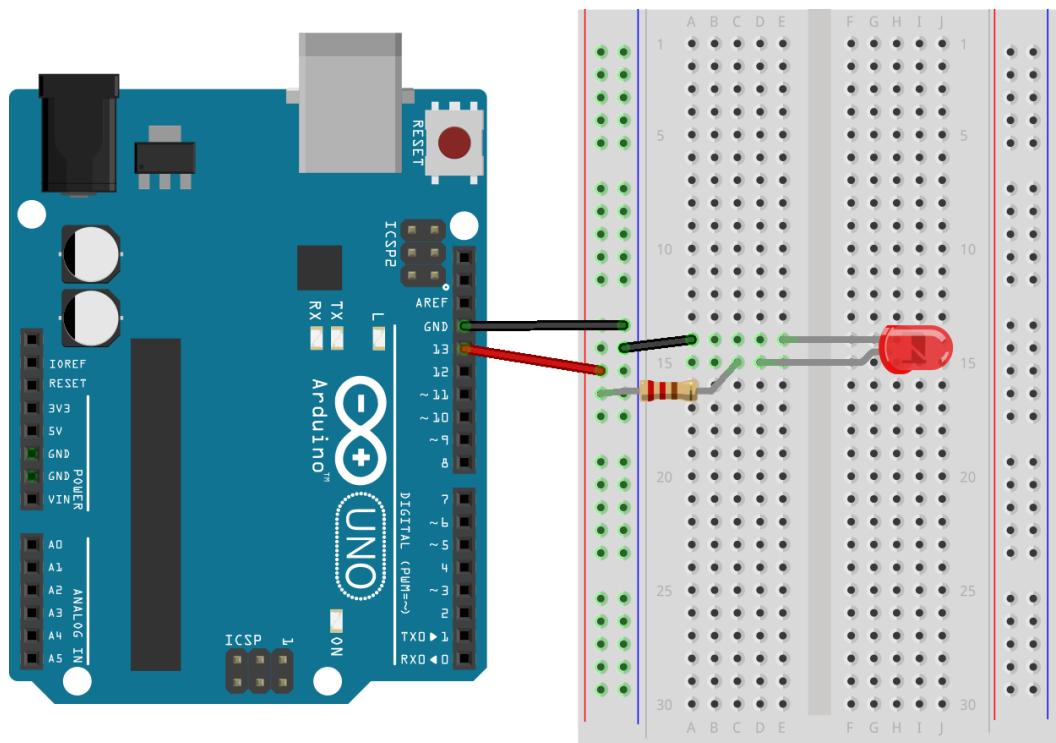
A JDĚTE NA TO ...

- ① Podle schématu zapojte elektronický obvod.



DEJTE SI POZOR

- Pozor si dejte na to, jak zapojujete LED diodu. Delší vývod musí být připojen přes rezistor k pinu. Kratší vývod je připojen na zem (pin GND).
- Dejte si pozor na hodnotu rezistoru. Zkontrolujte si, že je barevně označen v pořadí červená, červená, modrá, černá, zlatá.



② Spusťte program Arduino IDE a napište následující programový kód.

```
void setup() {
    pinMode(13, OUTPUT); // nastavení pinu 13 jako výstup
}

void loop() {
    digitalWrite(13, HIGH); // nastavení hodnoty HIGH, pro
                           // rozsvícení LED diody
    delay(1000);          // přerušení programu na 1 sekundu
    digitalWrite(13, LOW); // nastavení hodnoty LOW, pro zhasnutí
                           // LED diody
    delay(1000);
}
```

③ Po napsání programu připojte USB kabel k desce a k počítači.

④ V programu Arduino IDE nastavte odpovídající desku. V menu **Tools > Board > Arduino UNO**.

⑤ Dále nastavte port v Menu **Tools > Seriál Ports > vyberte odpovídající port**.

⑥ Pro nahrání programu do desky Arduino, klikněte na ikonu ➔



OTÁZKA

➔ Co se děje s diodou, pokud je program v pořádku nahrán?

Pokud vše funguje, tak výborně. Můžete se vrhnout na samostatné úkoly.

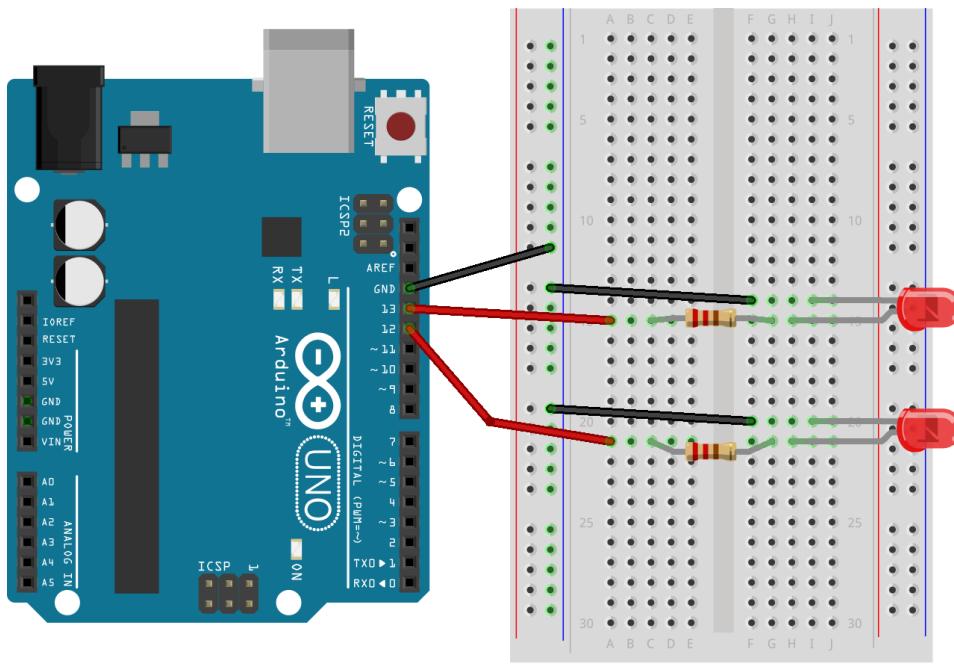
ÚKOLY PRO VÁS

- ➔ A) Připojte druhou LED diodu, třeba na pin 12.
- ➔ B) Sestavte program pro střídavé blikání obou LED diod, vždy po 1 sekundě.
- ➔ C) Sestavte program pro střídavé blikání LED diod, ale tentokrát každá blikne dvakrát, vždy po 1 sekundě. Mezi oběma diodami bude pauza dvě sekundy.
- ➔ D) Upravte program tak, aby diody blikaly postupně tj. rozsvítí se první, po 0,25 s se rozsvítí druhá, v čase 0,5 s zhasne první a v čase 0,75 s zhasne druhá. V čase 1 s rozsvítí první a pořád dokola.



ŘEŠENÍ ÚLOH

Úkol A)



Úkol B)

```
1 void setup() {
2     pinMode(13, OUTPUT);
3     pinMode(12, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(13, HIGH);
8     digitalWrite(12, LOW);
9     delay(1000);
10    digitalWrite(13, LOW);
11    digitalWrite(12, HIGH);
12    delay(1000);
13 }
```

Úkol C)

```
1 void setup() {
2     pinMode(13, OUTPUT);
3     pinMode(12, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(12, LOW);
8     digitalWrite(13, HIGH);
9     delay(1000);
10    digitalWrite(13, LOW);
11    delay(1000);
12    digitalWrite(13, HIGH);
13    delay(1000);
14    digitalWrite(13, LOW);
15    delay(2000);
16    digitalWrite(12, HIGH);
17    delay(1000);
18    digitalWrite(12, LOW);
19    delay(1000);
20    digitalWrite(12, HIGH);
21    delay(1000);
22    digitalWrite(12, LOW);
23    delay(2000);
24 }
```

Úkol D)

```
1 void setup() {
2     pinMode(13, OUTPUT);
3     pinMode(12, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(13, HIGH);
8     delay(250);
9     digitalWrite(12, HIGH);
10    delay(500);
11    digitalWrite(13, LOW);
12    delay(750);
13    digitalWrite(12, LOW);
14    delay(1000);
15 }
```

PRŮVODCE HODINOU II



Studenti budou pokračovat v programování elektronických obvodů. Tentokrát v zapojení sestaví základní obvod s využitím piezzo bzučák. Jedná se o volné pokračování předchozího zapojení s LED diodou. Studenti využijí nabyté znalosti získané při zapojování a programování elektronického obvodu s LED diodou. V programování využijí nové funkce určené pro ovládání bzučáku.



PŘÍPRAVA

Co bude v této hodině potřeba?

- ① Součásti obvodu – deska Arduino s USB kabelem, kontaktní pole, 1x piezzo bzučák, 1x rezistor 220Ω , 2x vodiče typu samec-samec.
- ② Osobní počítač pro studenty s nainstalovaným Arduino IDE.
- ③ Pokud je k dispozici, tak dataprojektor.
- ④ Prezentace k lekci 1, která je ke stažení na serveru GitHub
- ⑤ Pracovní listy pro studenty (ke stažení na GitHub).

1. KROK ⌚ 10 minut

Na úvod rozdejte studentům sady Arduino. Řekněte, že v této hodině naváží na předchozí příklad, který se týkal zapojení a ovládání obvodu s LED diodou. Naučí se programovat ovládání piezzo bzučáku zapojeného do obvodu, který je velmi podobný obvodu s LED diodou.



RYCHLÉ OPAKOVÁNÍ

- ➔ Zapojte obvod z předchozí hodiny, kde jste použili LED diodu.
- Naprogramujte tento obvod tak, že LED dioda bude blikat v intervalu 1 sekundy.



Opakování můžete urychlit tím, že studenti využijí vzorový příklad, dostupný v Arduino IDE. Stačí otevřít **File > Examples > Basic > Blink**. Otevře se základní programový kód.

2. KROK ⏳ 5 minut

Sestavení příkladu s LED diodou vede k tomu, že jej lze velmi snadno inovovat na obvod s piezzo bzučákem. Ať studenti vymění diodu za bzučák podle přiloženého schématu v pracovním listu nebo podle promítaného obrazu pomocí dataprojektoru z prezentace.

3. KROK ⏳ 10 minut

Ať si studenti spustí Arduino IDE a napíší základní program. Řekněte studentům, aby připojili USB kabel k desce a do počítače. Kliknutím na ikonu pro upload kódu ➔ ať nahrají program do desky Arduino.

JENOM PRO PŘIPOMENUTÍ

➔ Pokud nepůjde nahrát program do desky, studenti si musí zkontrolovat, zda mají v Arduino IDE vybranou odpovídající desku a komunikační port.



Studentům vysvětlete programový kód, zejména použité funkce týkající se bzučáku – **Tone**, **noTone**. Určitě nezapomeňte na deklaraci proměnné **bzucak**.



Při vysvětlování programového kódu zkuste formou otázek, zda studenti na funkcionality funkcí **Tone**, **noTone** přijdou sami.

ZEPTEJTE SE STUDENTŮ

- ➔ **Co se stane, když změníte číslo ve funkci Tone za proměnnou bzucak?**
Zvuk bzúčáku změní svou výšku.
- ➔ **Přidejte před funkci noTone dvě lomítka – symbol komentáře.**
- ➔ **Co se po přidání komentáře stane?**
Vyřadí se funkce noTone a bzúčák bude neustále pískat.



4. KROK 20 minut

Pokud studenti vše zvládli, mohou řešit **samostatně úkoly**. Tentokrát postupně použijí jak bzúčák, tak LED diodu.

ÚKOLY PRO STUDENTY



- ➔ **A) Napište program, který přehraje základní stupnici. Interval tónů je na vás.**
(Frekvence: 261, 294, 329, 349, 392, 440, 493, 523)
- ➔ **C) Změňte intervaly a pořadí tónů tak, aby vznikla melodie.**
- ➔ **B) Přidejte do obvodu na libovolný pin LED diodu. Dioda bude blikat v rytmu tónů.**



MOŽNÝ NÁPAD

- ➔ Studenti mohou realizovat závěrečný projekt, který je uveden jako „Mluvící robot“. Předpokladem je, že si z přiložené šablony doma vyrobí model robota, do kterého zabudují LED diodu a bzúčák.
- ➔ Robota naprogramují tak, aby mluvil vlastní, robotickou řečí. Inspirací může být robot R2D2 z Hvězdných válek.

PRACOVNÍ LIST – PIEZZO BZUČÁK

POKRAČOVÁNÍ V SEZNAMOVÁNÍ SE S PLATFORMOU ARDUINO. TENTOKRÁT BUDETE PROGRAMOVAT ELEKTRONICKÝ OBVOD, VE KTERÉM JE ZAPOJEN PIEZZO BZUČÁK A NÁSLEDNĚ PŘIDÁTE LED DIODU.

CO SE NAUČÍTE

- ① Zopakujete si, jak vytvořit elektronický obvod.
- ② Dokážete zapojit bzučák.
- ③ Vytvoření programu pro ovládání bzučáku.
- ④ Naprogramujete obvod s bzučákem a LED diodou.



CO BUDETE POTŘEBOVAT

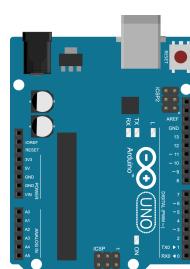
- ① Bzučák.
- ② Rezistor 220Ω.
- ③ Desku Arduino.
- ④ Kontaktní pole.
- ⑤ Vodiče typu samec-samec.



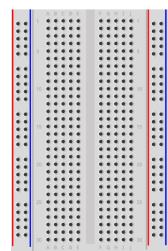
Piezzo bzučák



Rezistor 220Ω



Deska Arduino



Kontaktní pole

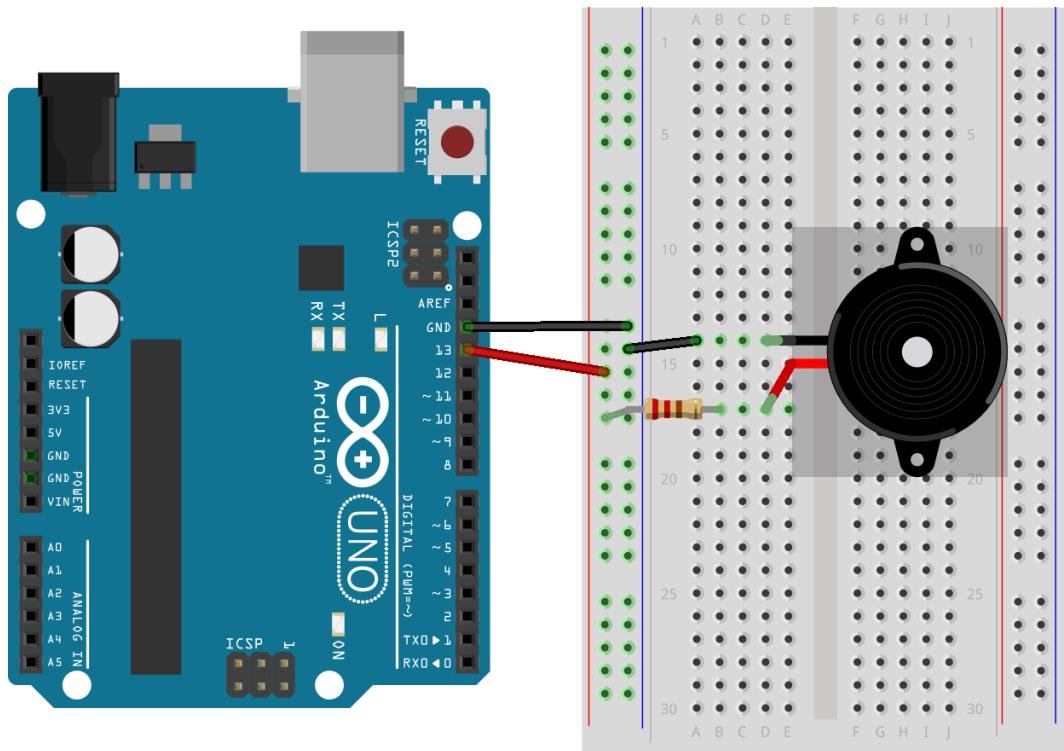
POUŽITÉ SOUČÁSTKY

A JDĚTE NA TO ...

- ① Podle schématu zapojte elektronický obvod.

DEJTE SI POZOR

- Pozor si dejte na to, jak zapojujete bzučák. Bzučák má na sobě nálepku s vyznačením kontaktu pro napájení, tj. symbol plus (+).
- Dejte si pozor na hodnotu rezistoru. Zkontrolujte si, že je barevně označen v pořadí červená, červená, modrá, černá, zlatá.



② Spusťte program Arduino IDE a napište následující programový kód.

```
const int bzucak=13; // definice proměnné bzucak

void setup() {
    pinMode(bzucak, OUTPUT); // nastavení pinu jako výstup
}

void loop() {
    tone(bzucak, 440); // funkce pro přehrání tónu
    delay(1000);
    noTone(bzucak); // funkce pro přerušení tónu
    delay(1000);
}
```

③ Po napsání programu připojte USB kabel k desce a k počítači.

- ④ V programu Arduino IDE nastavte odpovídající desku. V menu **Tools > Board > Arduino UNO**.
- ⑤ Dále nastavte port v Menu **Tools > Seriál Ports > vyberte odpovídající port**.
- ⑥ Pro nahrání programu do desky Arduino klikněte na ikonu ➔



OTÁZKA

- ➔ K čemu slouží v tomto příkladu funkce `delay(1000)`?
Pokud nevíte, tak vyzkoušejte změnit hodnotu parametru a spusťte program s touto novou hodnotou.

Pokud vše funguje, tak výborně. Můžete se vrhnout na samostatné úkoly.

ÚKOLY PRO VÁS



- ➔ A) Napište program, který přehraje základní stupnici. Interval tónů je na vás.
(Frekvence: 261, 294, 329, 349, 392, 440, 493, 523)
- ➔ B) Změňte intervaly a pořadí tónů tak, aby vznikla melodie.
- ➔ C) Přidejte do obvodu na libovolný pin LED diodu. Dioda bude blikat v rytmu tónů.

ŘEŠENÍ ÚLOH

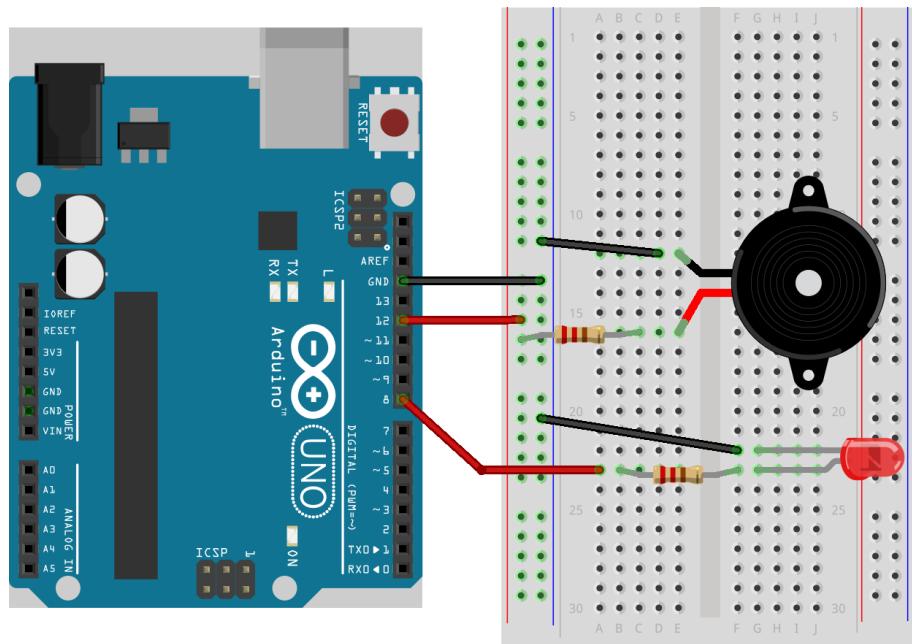
Úkol A)

```
1 const int pinBzucak=13;
2
3 void setup() {
4     pinMode(pinBzucak, OUTPUT);
5 }
6
7 void loop() {
8     // Ton C
9     tone(pinBzucak, 261);
10    delay(1000);
11    noTone(pinBzucak);
12    delay(1000);
13    // Ton D
14    tone(pinBzucak, 294);
15    delay(1000);
16    noTone(pinBzucak);
17    delay(1000);
18    // Ton E
19    tone(pinBzucak, 329);
20    delay(1000);
21    noTone(pinBzucak);
22    delay(1000);
23    // Ton F
24    tone(pinBzucak, 349);
25    delay(1000);
26    noTone(pinBzucak);
27    delay(1000);
28
29    // Další část kódu se neustále opakuje. Mění se pouze
30    // frekvence, které jsou parametrem funkce tone().
31
32 }
```

Úkol B)

```
1 // Uvedený kód ukazuje část melodie Jingle Bells.  
2 // Poradi tonu je: E E E P, E E E P, E G C D E P  
3 // Symbol P značí pauzu a bude mít hodnotu 0  
4  
5 const int pinBzucak=13;  
6  
7 void setup() {  
8     pinMode(pinBzucak, OUTPUT);  
9 }  
10  
11 void loop() {  
12     tone(pinBzucak, 329); delay(1000);  
13     noTone(pinBzucak); delay(100);  
14     tone(pinBzucak, 329); delay(1000);  
15     noTone(pinBzucak); delay(100);  
16     tone(pinBzucak, 329); delay(1000);  
17     noTone(pinBzucak); delay(100);  
18     tone(pinBzucak, 0); delay(1000);  
19  
20     noTone(pinBzucak); delay(100);  
21     tone(pinBzucak, 329); delay(1000);  
22     noTone(pinBzucak); delay(100);  
23     tone(pinBzucak, 329); delay(1000);  
24     noTone(pinBzucak); delay(100);  
25     tone(pinBzucak, 329); delay(1000);  
26     noTone(pinBzucak); delay(100);  
27     tone(pinBzucak, 0); delay(1000);  
28  
29     noTone(pinBzucak); delay(100);  
30     tone(pinBzucak, 329); delay(1000);  
31     noTone(pinBzucak); delay(100);  
32     tone(pinBzucak, 392); delay(1000);  
33     noTone(pinBzucak); delay(100);  
34     tone(pinBzucak, 261); delay(1000);  
35     noTone(pinBzucak); delay(100);  
36     tone(pinBzucak, 294); delay(1000);  
37     noTone(pinBzucak); delay(100);  
38     tone(pinBzucak, 329); delay(1000);  
39     tone(pinBzucak, 0); delay(1000);  
40 }  
41 }
```

Úkol C)



```
1 // Ukázka kódu pro mluvícího robota
2
3 const int pinLed=8;           // pin pro LED
4 const int pinBzucak=12;       // pin pro bzucak
5
6 void setup() {
7     pinMode(pinLed, OUTPUT);
8     pinMode(pinBzucak, OUTPUT);
9 }
10
11 void loop() {
12     tone(pinBzucak, 433);      // neustale se opakujici
13     digitalWrite(pinLed, HIGH); // blok kódu, kde se dale
14     delay(100);               // meni frekvence tonu
15     noTone(pinBzucak);        // a pauza mezi tony
16     digitalWrite(pinLed, LOW);
17     delay(100);
18
19     tone(pinBzucak, 1033);
20     digitalWrite(pinLed, HIGH);
21     delay(300);
22     noTone(pinBzucak);
23     digitalWrite(pinLed, LOW);
```

```
24     delay(300);
25
26     tone(pinBzucak, 600);
27     digitalWrite(pinLed, HIGH);
28     delay(200);
29     noTone(pinBzucak);
30     digitalWrite(pinLed, LOW);
31     delay(200);
32
33     tone(pinBzucak, 800);
34     digitalWrite(pinLed, HIGH);
35     delay(500);
36     noTone(pinBzucak);
37     digitalWrite(pinLed, LOW);
38     delay(500);
39
40 }
```

PODROBNÝ PRŮVODCE TEORIÍ

PODROBNĚ ROZEPSANÉ PŘÍKLADY S POPISEM FUNKCIONALIT OBVODŮ A PROGRAMOVÉHO KÓDU A ŘEŠENÍ ÚKOLŮ A MOŽNÝCH PROBLÉMŮ PŘI NEFUNKČNOSTI OBVODŮ.

OBSAH PRŮVODCE

- ① Digitální vstup a výstup s popisem funkcí pro jejich ovládání.
- ② Podrobný popis zapojení obvodu s LED diodou.
- ③ Základní seznámení s jazykem Wiring.
- ④ Popis rozhraní Arduino IDE pro nahrání kódu do desky.
- ⑤ Řešené problémy při zapojení LED diody.
- ⑥ Seznámení s programem Arduino IDE pro nahrávání kódu do desky.
- ⑦ Princip a zapojení piezzo bzučáku s podrobným vysvětlením.
- ⑧ Programový kód pro ovládání obvodu s bzučákem.
- ⑨ Technická část pro závěrečný projekt – Mluvící robot.
- ⑩ Vysvětlení řešení samostatných úkolů.

DIGITÁLNÍ VSTUP A VÝSTUP

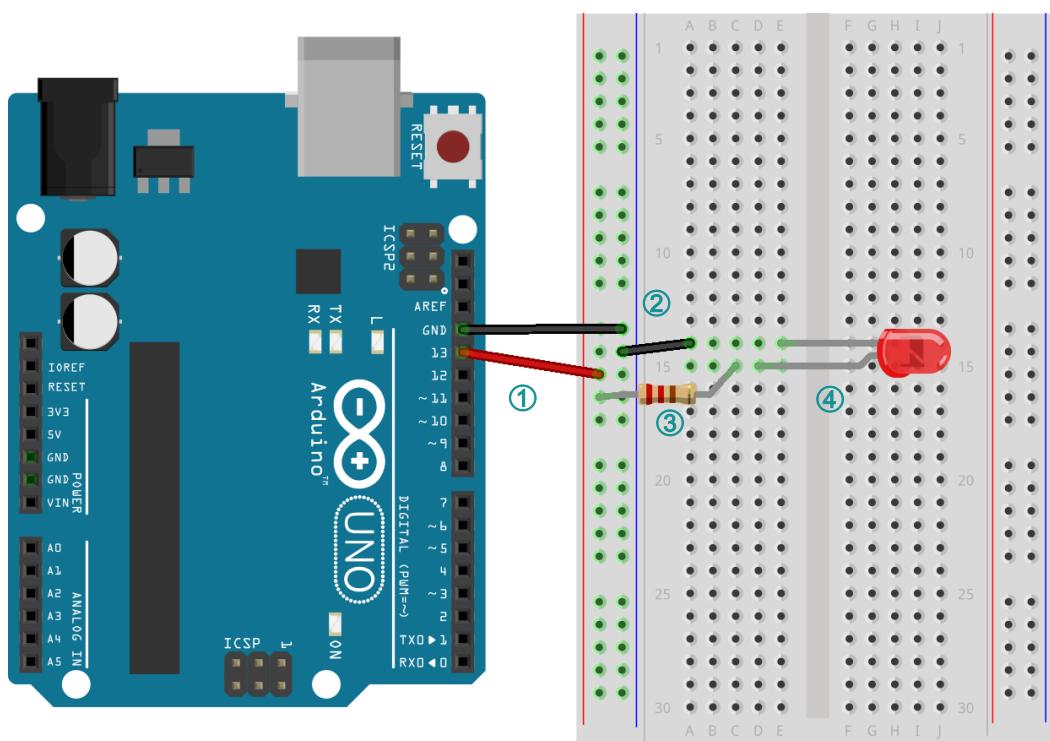
Téměř všechny piny Arduina lze použít jako digitální vstup nebo výstup. Jejich směr lze přepnout zavolením funkce **pinMode** a to v libovolném okamžiku. To znamená, že jeden pin může jednu chvíli sloužit jako vstup a poté i jako výstup. Jelikož se jedná o digitální piny, napětí na nich je na desce Arduino interpretováno pomocí dvou logických hodnot – **HIGH** a **LOW**. Často se tyto hodnoty interpretují také jako logická jedna a nula.

Logická hodnota **HIGH** je reprezentována napětím blízkým napájecímu, tzn. v případě desky Arduino UNO to je 5 V a logická hodnota **LOW** poskytuje napětí blízké 0 V.



Než si ukážeme konkrétní aplikaci programového kódu, vytvoříme elektronický obvod, na kterém bude vysvětlení digitálního vstupu a výstupu názornější.

ZAPOJENÍ OBVODU S LED DIODOU



Obr. 1 - Zapojení LED na digitální pin

- ① Zapojení obvodu je velmi jednoduché. Na digitální pin **13** je připojen červený vodič. Jeho druhý konec je připojen do kontaktního pole, do zdířky v části pro napájení.
- ② Černý vodič je veden z pinu země **GND** a připojen na kontaktní pole. Dále je připojen k diodě.
- ③ Aby nedošlo k rychlému zničení LED diody, je jí předřazen rezistor **220Ω**. Jedna část rezistoru je zapojena do zdířky kontaktního pole, kde je připojen pin **13** a druhá k anodě LED diody.
- ④ Samotná dioda je připojena do kontaktního pole tak, že anoda (delší kontakt) je spojena s rezistorem a katoda se zemí (černý vodič).



Všimněte si, že obvod je uzavřený. Nikde není mechanický prvek, který by ho rozpojoval tak, jak bylo uvedeno v zapojení znázorňující sériový a paralelní obvod. Ovládání diody bude v tomto obvodu zajišťovat programový kód.



KRÁTCHE O JAZYKU WIRING

Arduino je možné programovat v jazyce C nebo C++. Nejjednodušší je však používat knihovnu Wiring. Ta je v současné době pro programování Arduina velmi rozšířená. Kvůli její komplexnosti se o ní mluví jako o samostatném programovacím jazyku.

Program pro Arduino v jazyce Wiring se nazývá sketch [skeč] a skládá se z několika základních částí:

- ① deklarace tzv. globálních proměnných, definice konstant a načtení externích knihoven,
- ② definice funkce **setup()**,
- ③ definice funkce **loop()**,
- ④ definice zbývajících funkcí programu.

Podrobné informace a referenční příručku jazyka Wiring najeznete v kapitole **Programovací jazyk Wiring**.

PROGRAMOVÝ KÓD

K vytváření skečů v programování v jazyce Wiring můžeme použít jakýkoliv textový editor. My ale využijeme software Arduino IDE, který poskytuje kromě možnosti psaní programového kódu i nahrání do desky Arduino. Popis hlavních funkcí softwaru Arduino IDE je uveden v kapitole [Upload programu do desky](#).

ZÁKLADNÍ PŘÍKLAD

Zde je uveden základní programový kód, který rozbliká LED diodu.

```
1 void setup() {  
2     pinMode(13, OUTPUT);  
3 }  
4  
5 void loop() {  
6     digitalWrite(13, HIGH);  
7     delay(1000);  
8     digitalWrite(13, LOW);  
9     delay(1000);  
10 }  
11 }
```

The diagram consists of seven numbered callouts (1 through 7) pointing from the right side to specific lines of code on the left. Callout 1 points to the start of the `setup()` function. Callout 2 points to the `pinMode` function call within `setup()`. Callout 3 points to the start of the `loop()` function. Callout 4 points to the `digitalWrite` function call within `loop()` that sets the pin high. Callout 5 points to the `delay` function call within `loop()` that waits for 1000 milliseconds. Callout 6 points to the `digitalWrite` function call within `loop()` that sets the pin low. Callout 7 points to the second `delay` function call within `loop()` that waits for another 1000 milliseconds.

- ① Funkce `setup()` je inicializační a vykoná se pouze jednou při způštění programu.
- ② Funkce `pinMode()` nastaví pin 13 jako výstup `OUTPUT`.
- ③ Funkce `loop()` je výkonná a je volána následně po funkci `setup()`. Vykonává se stále dokola.
- ④ Funkce `digitalWrite()` rozsvítí LED diodu připojenou na pin **13** tak, že přepne výstup na hodnotu `HIGH`. Tím se pošle na uvedený pin 5V, což zajistí rozsvícení diody.
- ⑤ Funkce `delay()` přeruší běh programu na dobu uvedenou v závorce. Hodnota se zadává v milisekundách, tj. 1000ms = 1s.
- ⑥ Opět zavoláme funkci `digitalWrite()`, tentokrát však s hodnotou `LOW`, což znamená, že na pinu **13** bude napětí změněno na 0V. Tím dojde k zhasnutí LED diody.
- ⑦ Následuje opět funkce pro pozastavení běhu programu na 1 sekundu.

A to je celý program. Mimo základního kódu ve funkcích **setup()** a **loop()** tento program neobsahuje žádné deklarace, knihovny či jiné funkce. Protože veškerý výkonný kód je uveden ve funkci **loop()**, která ho vykonává pořád dokola, dioda bude neustále blikat.

UPLOAD PROGRAMU DO DESKY

Program již sice máme napsaný a víme co má dělat, ale ještě není nahrán v desce Arduino. K tomu využijeme stejný software, jaký jsme použili pro psaní kódu.

- ① Klikněte na ikonu pro ověření kódu. Mělo by se objevit něco jako **Binary sketch size: 934 bytes (of a 30720 byte maximum)**. Tedy že skeč přeložený do binární podoby zabírá 934 bajtů.
- ② Pokud máte přeloženo, klikněte na ikonu pro nahrání do desky. IDE přeloží skeč a nahraje do Arduina. Počká pár sekund a pošle signál RESET. Dioda by se měla rozblíkat.



Ověřit

Ověří, zda napsaný kód je v pořádku a zkompiluje ho.



Nahrát

Zkompiluje kód, nahraje jej do nakonfigurované desky.



Nový

Vytvoří nový skeč.



Uložit

Uloží skeč.



Sériový monitor

Otevře sériový monitor.



NELZE NAHRÁT KÓD DO DESKY

USB kabel – ujistěte se, že máte desku Arduino připojenou k počítači.

Chyba v kódu – zkонтrolujte, jestli je programový kód opravdu správně napsán.

Pokud bude existovat syntaktická chyba, kód se do desky nenahraje.

Správná deska – přesvědčte se, že máte správně zvolenou desku v nabídkách **Tools>Board**.

Správný port – před nahráním náčrtu je třeba vybrat správný port v nabídkách

Tools>Ports.

DIODA NEBLIKÁ

Zapojení diody – zkонтrolujte, zda je LED dioda správně zapojená, tzn. delší vývod z diody (anoda) je připojen přes rezistor na pin 13.

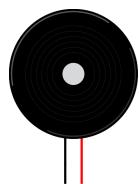
Rezistor – zkонтrolujte hodnotu rezistoru, zda není příliš veliká.



(Př. 1) Připojte ještě jednu LED diodu. Použijte k tomu libovolný pin. Upravte skeč tak, aby diody střídavě blikaly.

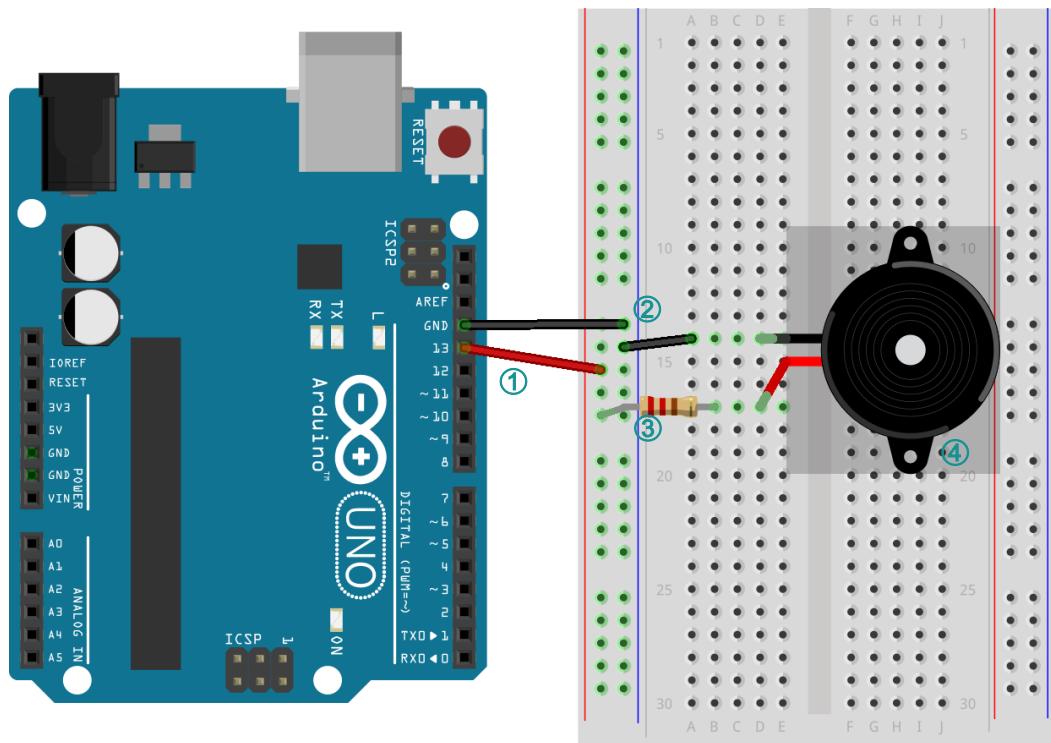
ZAPOJENÍ OBVODU S BZUČÁKEM

Tento příklad je lehkou inovací předchozího zapojení LED diody. Využívá se zde všech předchozích informací, pouze se použije zařízení, které přeměňuje elektrickou energii na zvuk. Tímto zařízením je bzučák.



Co je piezzo bzučák?

Piezzo bzučák je malé zařízení, které se rozvibruje, pokud se do něj přivede elektrické napětí. Tím se také rozvibrují částice v okolí a vzniká tak zvuková vlna.



Obr. 2 - Zapojení bzučáku

- ① Na digitální pin **13** je připojen červený vodič. Jeho druhý konec je připojen do kontaktního pole, do zdírky v části pro napájení.
- ② Černý vodič je veden z pinu země **GND** a připojen na kontaktní pole. Dále je vodič připojen k bzučáku.

- ③ Aby nedošlo ke zničení bzučáku, je předřazen rezistor **220Ω**. Jedna část rezistoru je zapojena do zdírky kontaktního pole, kde je připojen pin **13** a druhá ke kontaktu bzučáku, který je označen znaménkem **+**.
- ④ Bzučák je připojen do kontaktního pole tak, že kontakt označený znaménkem **+** je spojen s rezistorem a druhý kontakt se zemí (černý vodič).

PROGRAMOVÝ KÓD

ZÁKLADNÍ PŘÍKLAD

Příklad je ukázkou využití základních funkcí pro práci s bzučákem.

```
1  const int bzucak=13;           | ①
2
3
4  void setup() {
5      pinMode(bzucak, OUTPUT);   | ②
6  }
7
8  void loop() {
9      tone(bzucak, 440);        | ③
10     delay(1000);
11     noTone(bzucak);          | ④
12     delay(1000);
13 }
```

- ① Deklarace proměnné **bzucak**, která obsahuje číslo pinu, na který je připojen kladný vodič bzučáku.
- ② Definování zvoleného pinu jako výstupního.
- ③ Funkce **tone()** hraje tón na zadaném čísle pinu o dané frekvenci.
- ④ Tón hraje tak dlouho, dokud trvá průběh funkce **delay()**. Tón bude přerušen funkcí **noTone()**, která má jediný parametr a tím je číslo pinu.

Předchozí příklad lze ještě zjednodušit tak, že přidáme třetí parametr do funkce **tone()**. Tímto parametrem je délka trvání tónu.

```
1 const int bzucak=13;           └─①
2
3
4 void setup() {
5   pinMode(bzucak, OUTPUT);
6   tone(bzucak, 440, 2000);     └─②
7 }
8
9 void loop() {                  └─③
10 }
```

- ① Deklarace proměnné **bzucak**, která obsahuje číslo pinu, na který je připojen kladný vodič bzučáku.
- ② Definování zvoleného pinu jako výstupního.
- ③ Funkce **tone()** hraje tón a v uvedeném příkladu se využívá třech vstupních parametrů: číslo pinu, frekvence tónu, doba přehrávání. Tato funkce je záměrně napsána v části **setup()**. Pokud by byla v **loop()**, pak by se tón přehrál neustále dokola.



(Př. 2) Zapojte obvod s diodou a piezzo bzučákem společně. Napište program, který bude představovat mluvu robota. Robot bude vydávat zvuky podle vlastního azyka a zároveň mu bude blikat nos.



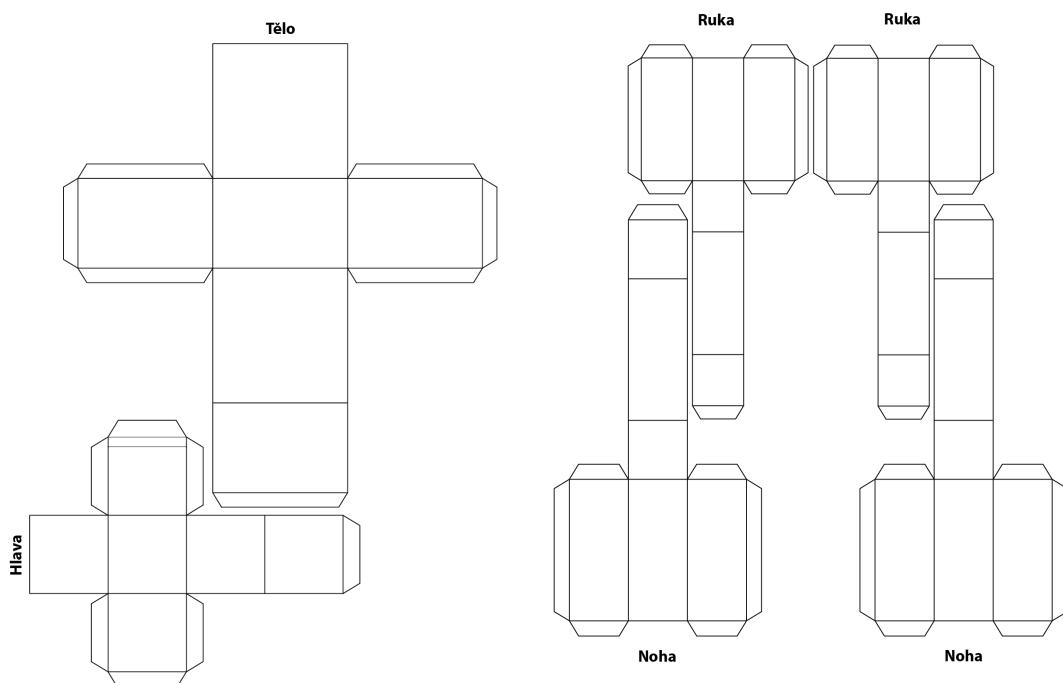
Po zapojení a naprogramování zadaného úkolu sestrojte robota podle přiloženého návodu. Do něj zabudujte diodu a bzučák.

MLUVÍCÍ ROBOT

Pro vytvoření robota budeme potřebovat: karton (tvrdší papír), lepidlo, nůžky.

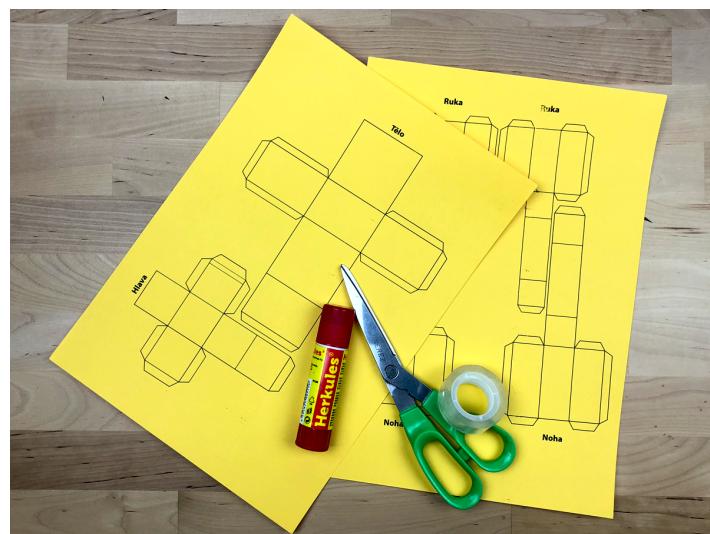
PAPÍROVÁ KONSTRUKCE

Robota si stáhněte a vytiskněte na karton. Můžete si vybarvit nebo vytvořit zcela vlastního robota podle své fantazie.



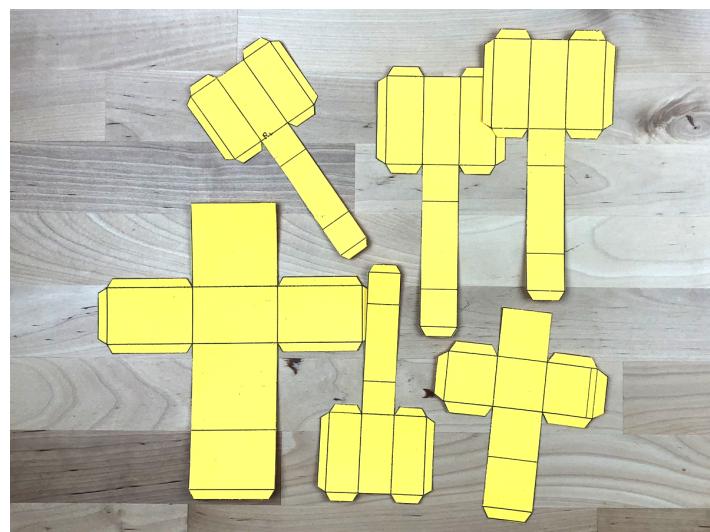
Obr. 3 - Šablona robota

Podle obrázku Obr. 3 - Šablona robota je vidět, že se robot skládá z několika kvádrů. Šablonu si stáhněte z adresy <https://github.com/Nowis75/PRIM/raw/master/sablona-robot.pdf>.



Obr. 4 - Vytiskněná šablona

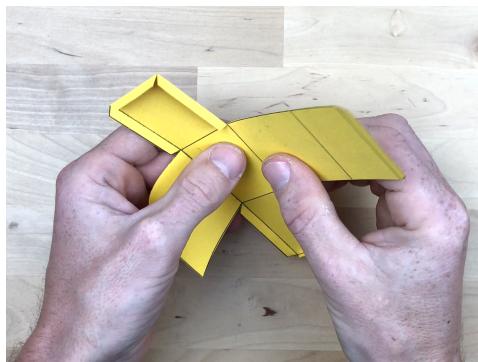
Podle plných čar vystříhněte vytiskněnou šablonu.



Obr. 5 - Vystřížené části robota

Jednotlivé kvádry složte tak, že naohýbáte jednotlivé hrany podle přerušovaných čar.

Lepidlo nanesete na spojnicové spoje.



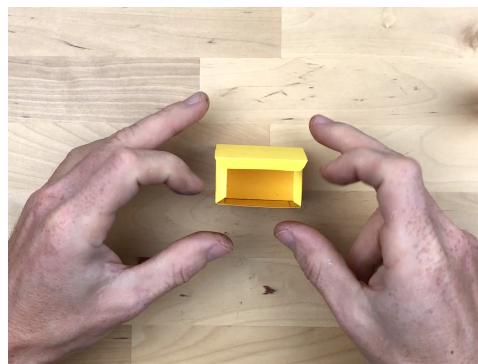
Obr. 6 – Ohýbání lepících hran



Obr. 7 – Lepení kvádru

Všechny vystřížené tvary složte a slepte. U hlavy a těla nechte horní část kvádru (víko) nezalepné.

Do hlavy robota nainstalujte LED diodu. Udělejte otvor pro diodu a pro vodiče. Pro připojení LED diody použijte vodiče samec > samice.



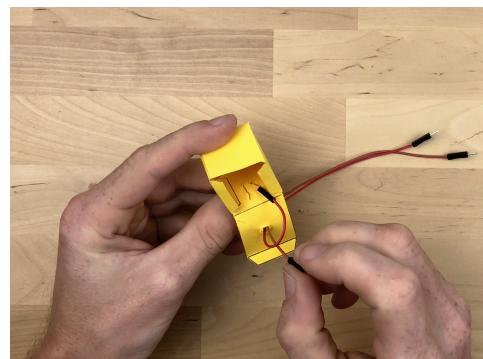
Obr. 8 – Slepěný kvádr



Obr. 9 – Robot z kvádrů

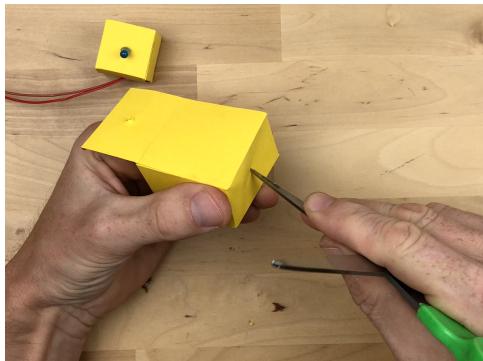


Obr. 10 – Instalace vodičů

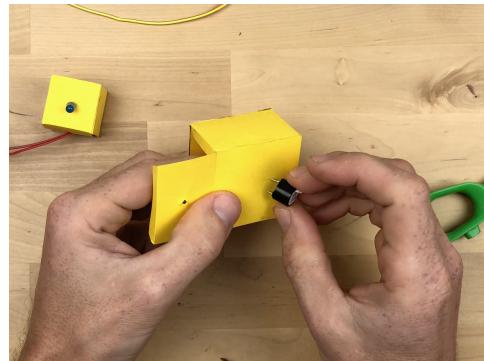


Obr. 11 – Instalace vodičů

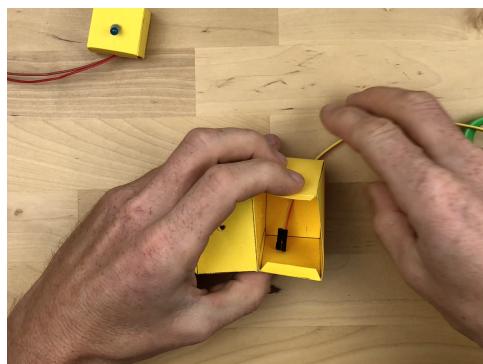
Do těla robota nainstalujte piezzo bzučák. Bzučák stačí připevnit na zadní část těla a dovnitř přivést dva vodiče typu samec > samice.



Obr. 12 – Tělo robota



Obr. 13 – Instalace piezzo bzučáku

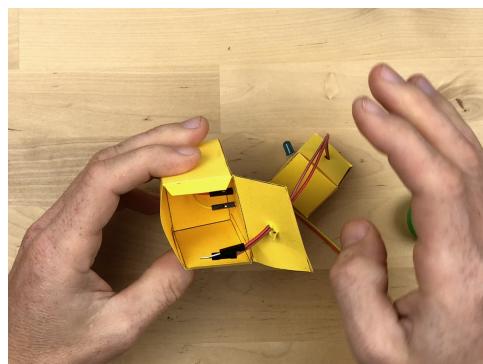


Obr. 14 – Instalace vodičů

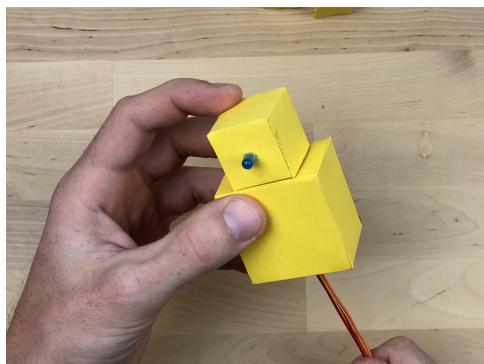


Obr. 15 – Připevnění bzučáku vodiči

Hlavu spojíme s tělem tak, že protáhneme vodiče víkem těla a jeho spodní částí. Z těla budou vycházet čtyři dráty.

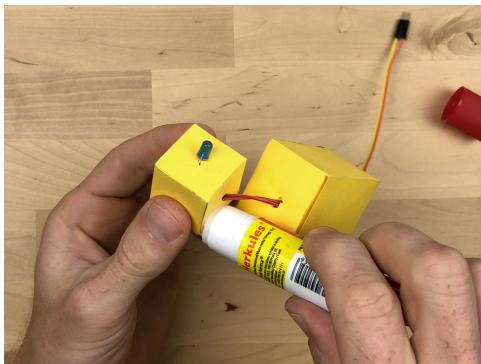


Obr. 16 – Instalace vodičů

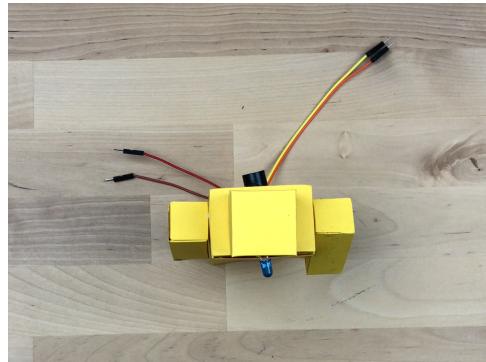


Obr. 17 – Připevnění bzučáku vodiči

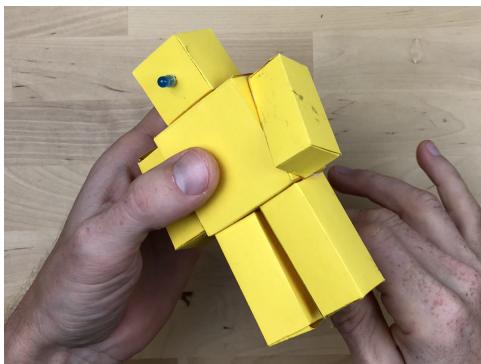
Nyní se přilepí hlava k tělu. Ostatní části budou přilepeny také k tělu, Obr. 20 – Sestavený robot.



Obr. 18 – Lepení hlavy k tělu



Obr. 19 – Pohled na robota ze shora

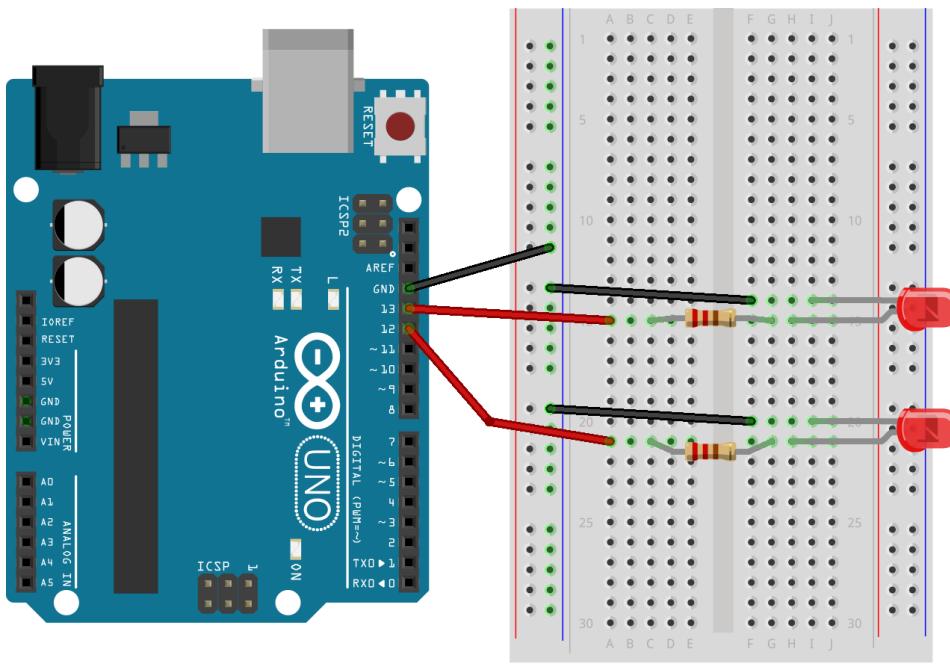


Obr. 20 – Sestavený robot

ŘEŠENÍ ÚLOH

(PŘ. 1

Řešení je velmi jednoduché. Je důležité, aby obě diody měly své napájení. Pro jednu z pinu 13 a druhou z pinu 12. Rezistory jsou zapojeny přes oddělovací polovinu kontaktní desky. Vodič země je přiveden do společné části (modrá čára) kontaktní desky a dále rozveden pro každou diodu zvlášť.

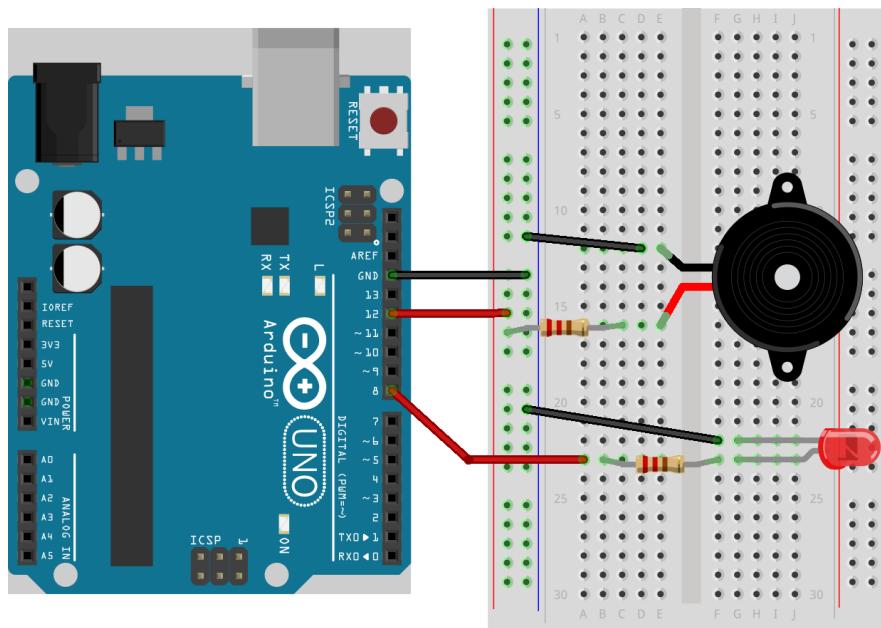


Programový kód je rozšířen o definici druhého výstupního pinu a následné střídání hodnot **HIGH** a **LOW** pro každou diodu pomocí funkce **digitalWrite()**.

```
1 void setup() {
2     pinMode(13, OUTPUT);
3     pinMode(12, OUTPUT);
4 }
5
6 void loop() {
7     digitalWrite(13, HIGH);
8     digitalWrite(12, LOW);
9     delay(1000);
10    digitalWrite(13, LOW);
11    digitalWrite(12, HIGH);
12    delay(1000);
13 }
```

(PŘ. 2

Příklad je spojením úloh, které se zabývaly LED diodou a bzučákem odděleně. Stačí správně nadefinovat výstupní piny.



V programovém kódu se opakovaně využívá volání funkce **digitalWrite()** pro rozsvěcování a zhasínání diody. Pro zvuk se využívá funkce **tone()**. Důležité je pořadí funkcí a přerušení programu pomocí funkce **delay()**.

```
1 // Ukázka kódu pro mluvícího robota
2
3 const int pinLed=12;           // pin pro LED
4 const int pinBzucak=8;        // pin pro bzucak
5
6 void setup() {
7     pinMode(pinLed, OUTPUT);
8     pinMode(pinBzucak, OUTPUT);
9 }
10
11 void loop() {
12     tone(pinBzucak, 433);      // neustale se opakujici
13     digitalWrite(pinLed, HIGH); // blok kódu, kde se dale
14     delay(100);               // meni frekvence tonu
15     noTone(pinBzucak);        // a pauza mezi tony
```

```
    digitalWrite(pinLed, LOW);    //  
    delay(100);                 //  
  
    tone(pinBzucak, 1033);  
    digitalWrite(pinLed, HIGH);  
    delay(300);  
    noTone(pinBzucak);  
    digitalWrite(pinLed, LOW);  
    delay(300);  
  
    tone(pinBzucak, 600);  
    digitalWrite(pinLed, HIGH);  
    delay(200);  
    noTone(pinBzucak);  
    digitalWrite(pinLed, LOW);  
    delay(200);  
  
    tone(pinBzucak, 800);  
    digitalWrite(pinLed, HIGH);  
    delay(500);  
    noTone(pinBzucak);  
    digitalWrite(pinLed, LOW);  
    delay(500);  
  
}
```