

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

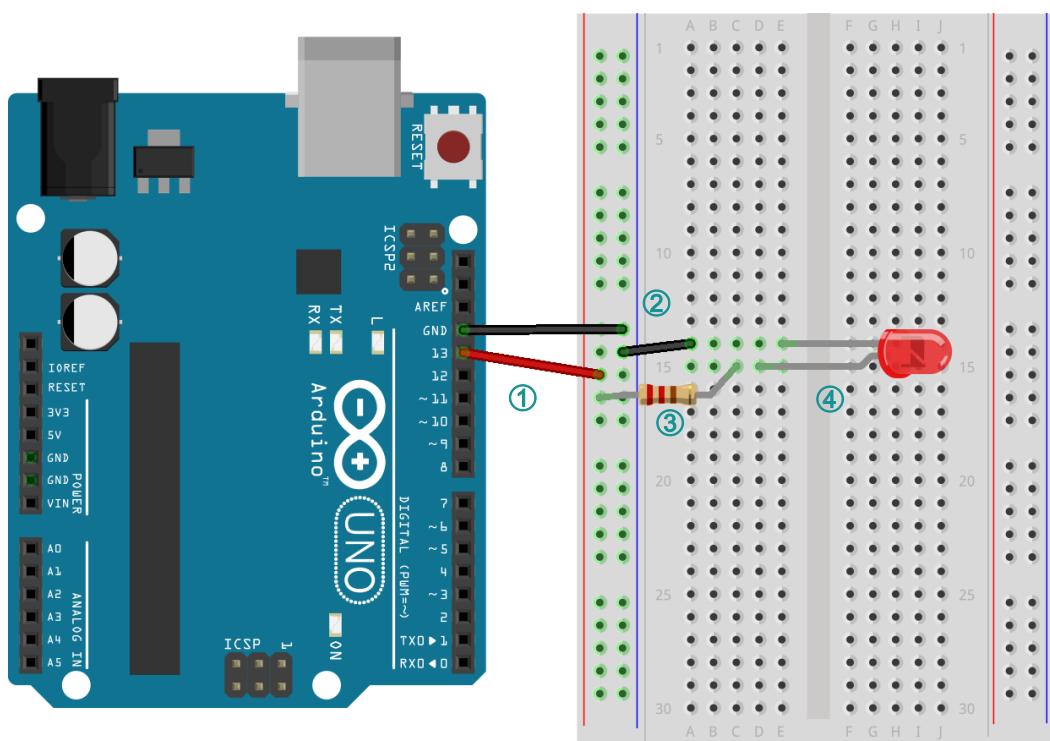
Téměř všechny piny Arduina lze použít jako digitální vstup nebo výstup. Jejich směr lze přepnout zavolením funkce **pinMode** a to v libovolném okamžiku. To znamená, že jeden pin může jednu chvíli sloužit jako vstup a poté i jako výstup. Jelikož se jedná o digitální piny, napětí na nich je na desce Arduino 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 }
```



- 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.

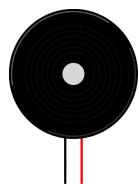
Rezistor – zkонтrolujte hodnotu rezistoru, zda není příliš veliká.



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

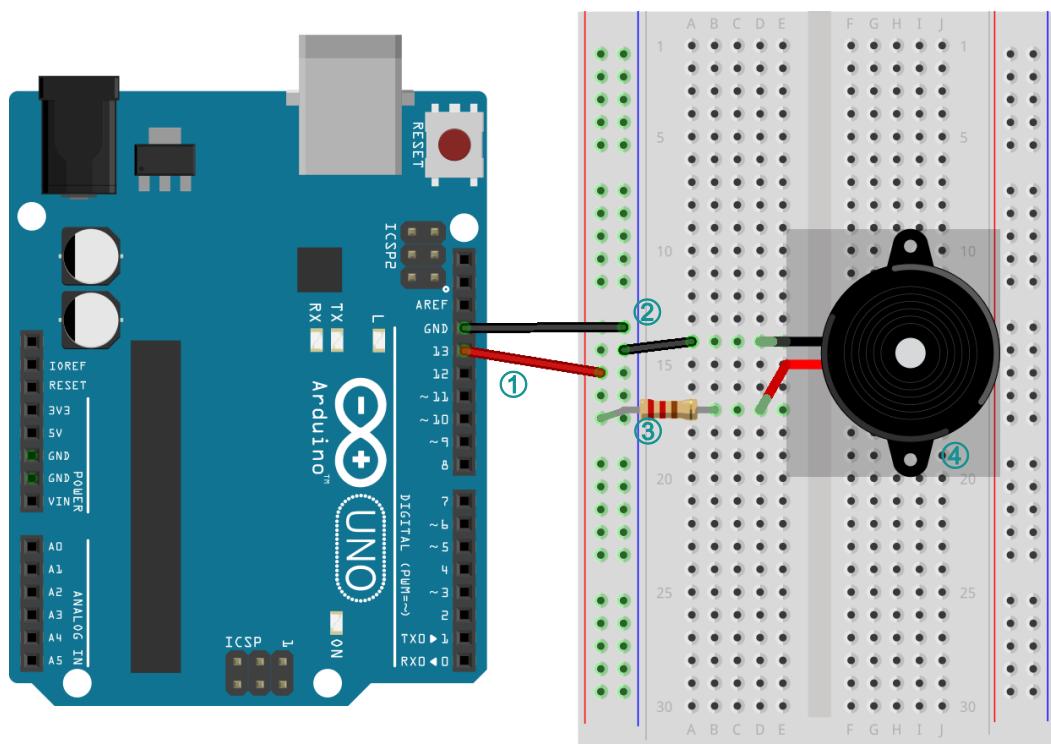
ZAPOJENÍ OBVODU S BZUČÁKEM

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 číslu 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 void setup() {
4     pinMode(bzucak, OUTPUT);
5     tone(bzucak, 440, 2000);                         ②
6 }
7
8 void loop() {                                       ③
9 }
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 jazyka a zároveň mu bude blikat nos.



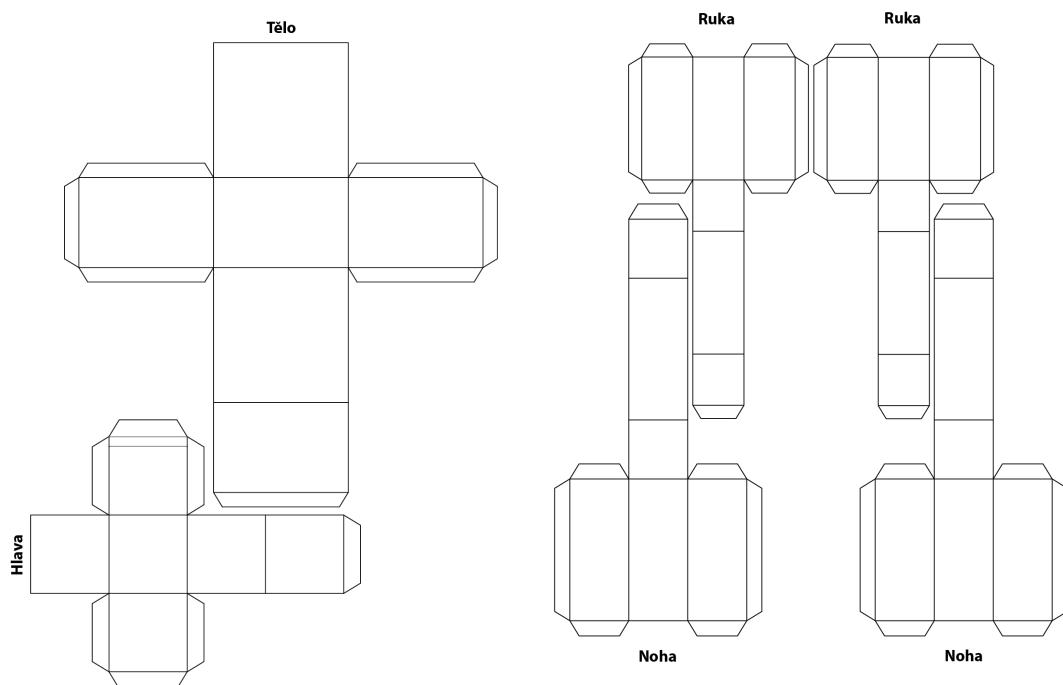
Po zapojení a naprogramování zadанého úkolu sestrojte robota podle přiloženého návodu. Do něj zabudujte diodu a bzučák.

MLUVÍCÍ ROBOT

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

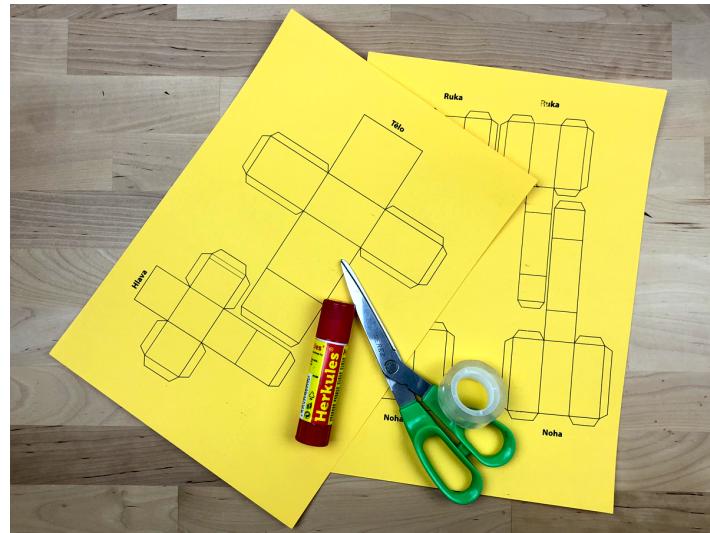
PAPÍROVÁ KONSTRUKCE

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



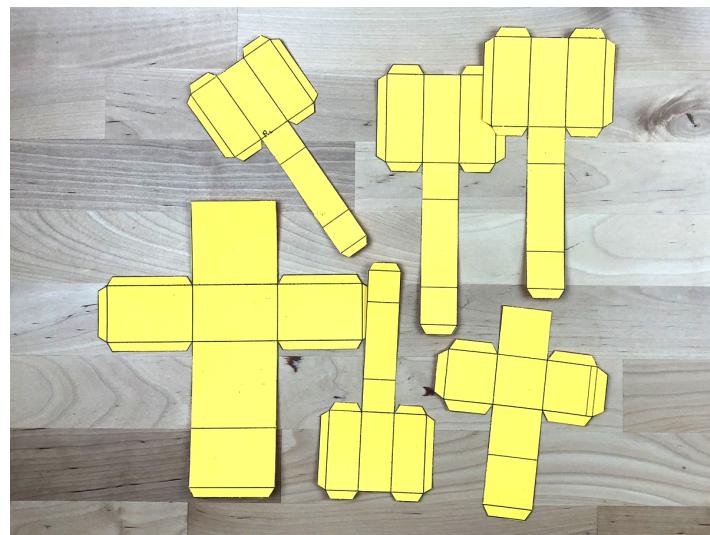
Obr. 3 - Šablona robota

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>.



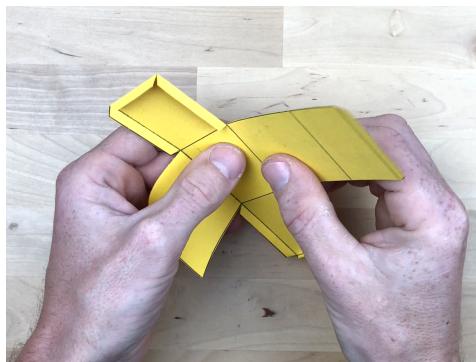
Obr. 4 - Vytisknuta šablona

Podle plných čar vystříhněte vytisknou šablonu.



Obr. 5 - Vystřížené části robota

Jednotlivé kvádry složte tak, že naohýbáte jednotlivé hrany podle přerušovaných čar.
Lepidlo nanesete na spojnicové spoje.



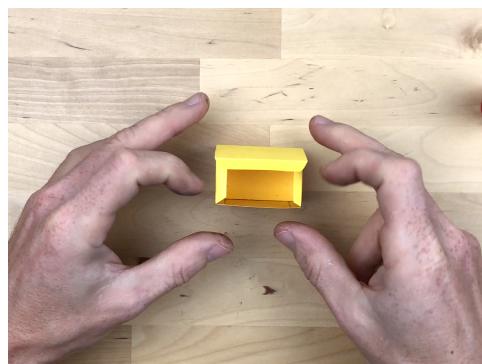
Obr. 6 – Ohýbání lepících hran



Obr. 7 – Lepení kvádru

Všechny vyštízené 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 zástrčka > samice.



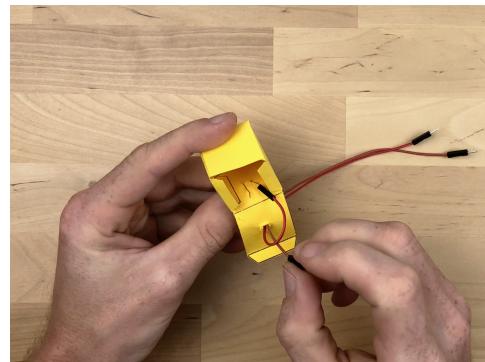
Obr. 8 – Slepéný kvádr



Obr. 9 – Robot z kvádrů

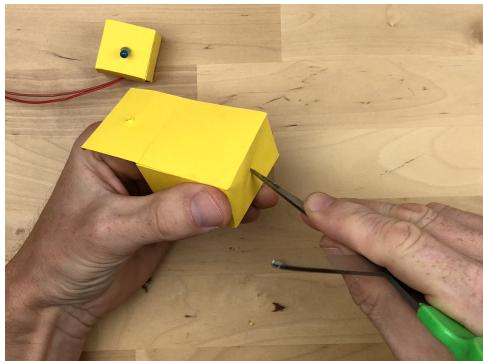


Obr. 10 – Instalace vodičů

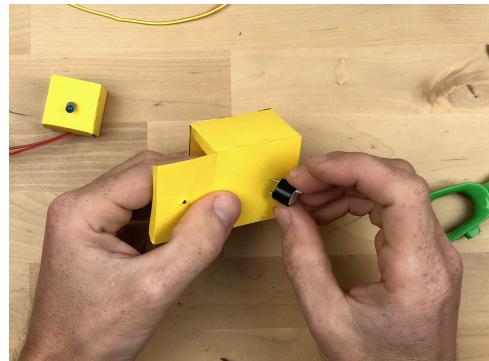


Obr. 11 – Instalace vodičů

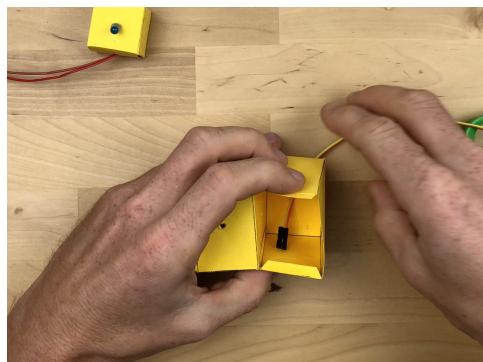
Do těla robota nainstalujte piezzo bzučák. Bzučák stačí připevnit na zadní část těla a dovnitř přivést dva vodiče typu zástrčka > samice.



Obr. 12 – Tělo robota



Obr. 13 – Instalace piezzo bzučáku

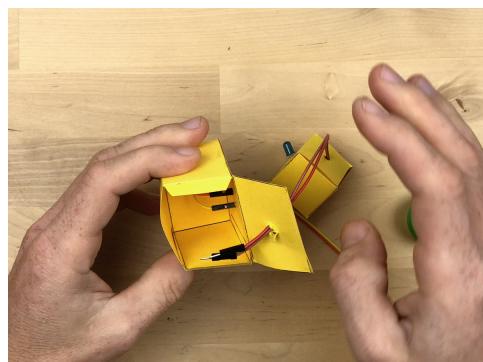


Obr. 14 – Instalace vodičů

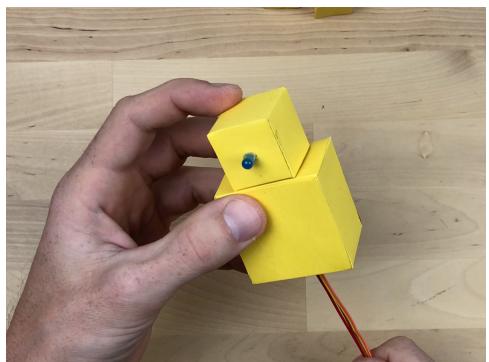


Obr. 15 – Připevnění bzučáku vodiči

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

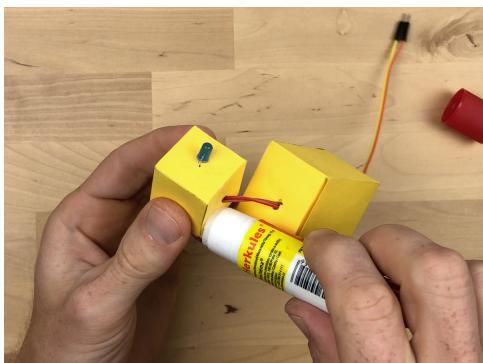


Obr. 16 – Instalace vodičů

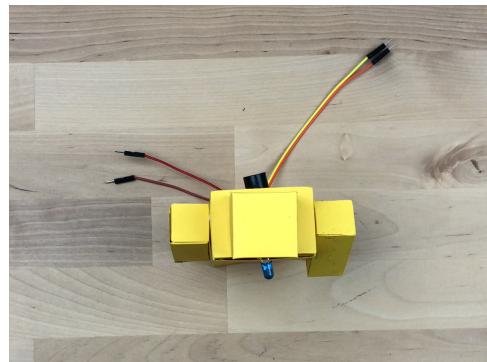


Obr. 17 – Připevnění bzučáku vodiči

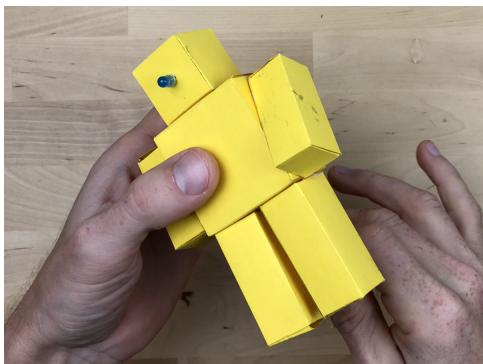
Nyní se přilepí hlava k tělu. Ostatní části budou přilepeny také k tělu, Obr. 20 – Sestavený robot.



Obr. 18 – Lepení hlavy k tělu



Obr. 19 – Pohled na robota ze shora

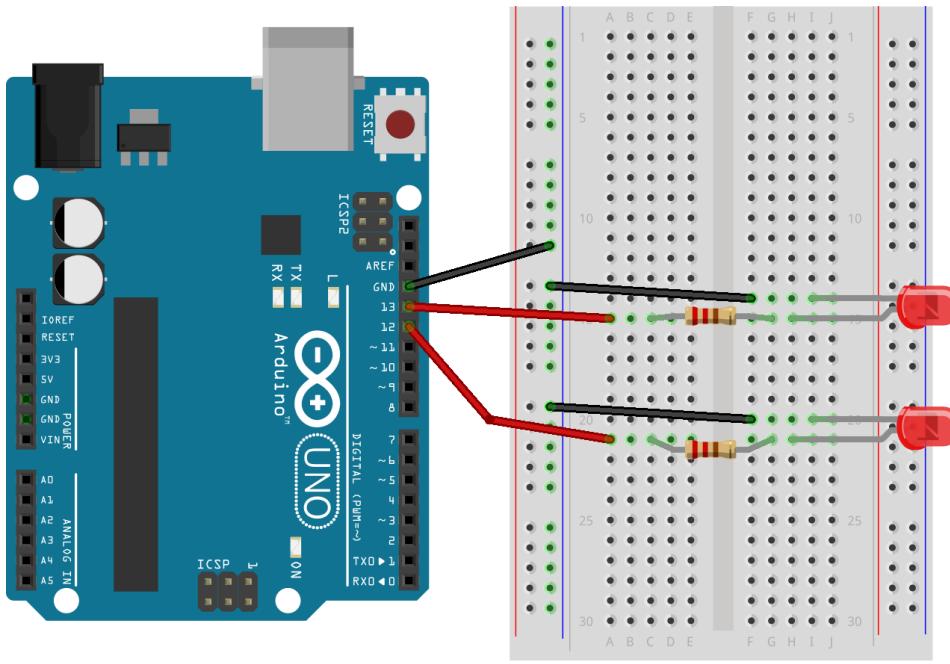


Obr. 20 – Sestavený robot

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.

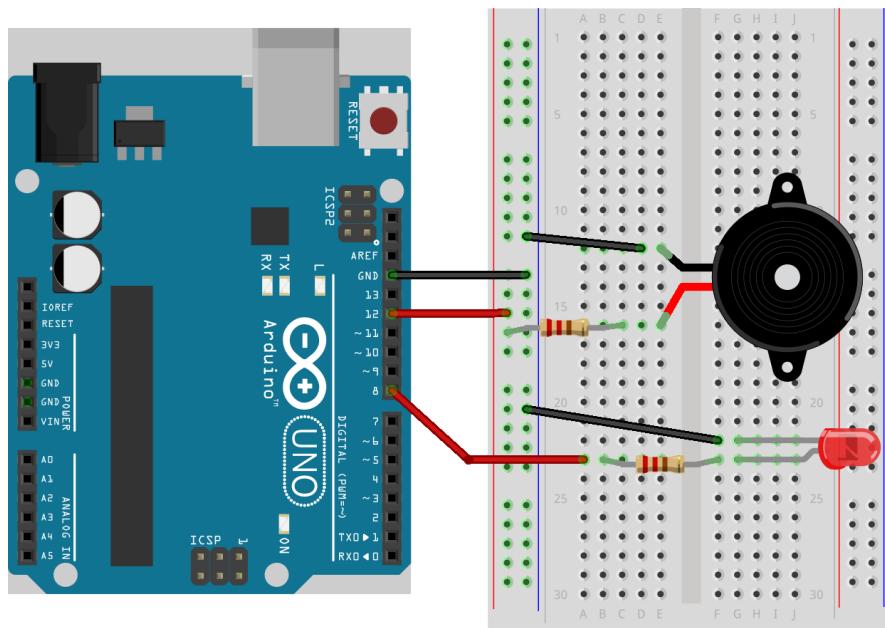


Programový kód je rozšířen o definici druhého výstupního pinu a následné střídání hodnot **HIGH** a **LOW** pro každou diodu pomocí funkce **digitalWrite()**.

```
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
                                // meni frekvence tonu
14   delay(100);                // a pauza mezi tony
15   noTone(pinBzucak);
```

```
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);  
  
}
```