

11. ULTRAZVUKOVÉ ČIDLO

JEDNÍM Z MNOHA SENZORŮ V ROBOTICE JE ULTRAZVUKOVÉ ČIDLO, KTERÉ NALEZNE UPLATNĚNÍ V CELÉ ŘADĚ APLIKACÍ. JEHO PRINCIP JE POMĚRNĚ JEDNODUCHÝ A TVOŘÍ NEDOMYSLITELNOU ČÁST SENZORICKÉHO SUBSYSTÉMU.

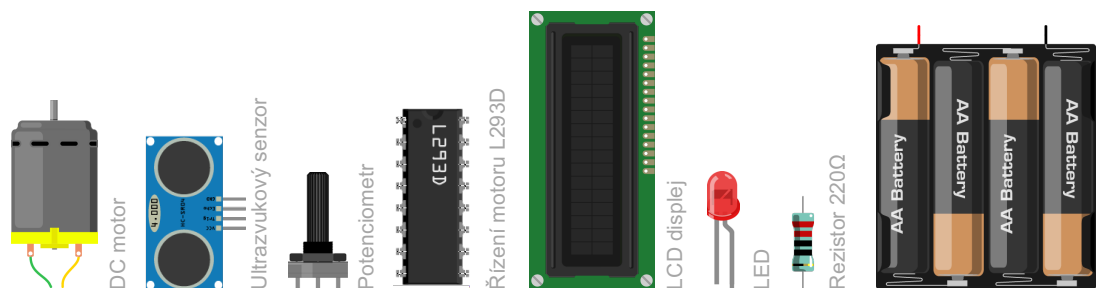
CÍLE

- ① Pochopení principu ultrazvukového senzoru.
- ② Zapojení ultrazvukového senzoru.
- ③ Využití externí knihovny pro ultrazvukový senzor.
- ④ Praktické příklady pro autonomii robotických prostředků.
- ⑤ Projekt robotický prostředek.

Čas: **4x45 min**

Úroveň: ■ ■ ■ ■ ■

Vychází z: **5, 6**



POUŽITÉ SOUČÁSTKY

PRŮVODCE HODINOU I



Žáci sestaví obvod, ve kterém bude zapojen ultrazvukový senzor. Na tomto obvodu jim bude vysvětlen princip tohoto senzoru a jeho programování. Tento obvod dále rozšíří o logickou podmínku pro testování vzdálenosti objektů před senzorem. Bude využito dosavadních poznatků z předchozích kapitol a to zejména v oblasti čtení z analogových vstupů.



PŘÍPRAVA

Co bude v této hodině potřeba?

- ⑥ Součásti obvodu – deska Arduino s USB kabelem, kontaktní pole, ultrazvukový senzor, vodiče typu zástrčka-zástrčka.
- ⑦ Osobní počítač pro studenty s nainstalovaným Arduino IDE.
- ⑧ Pokud je k dispozici, tak dataprojektor.
- ⑨ Prezentace k lekci 11.
- ⑩ Pracovní listy pro studenty.

1. KROK 🕒 10 minut

Na úvod rozdejte studentům sady Arduino. Řekněte, že náplní vašeho kurzu bude se naučit pracovat s ultrazvukovým senzorem.

ZEPTEJTE SE STUDENTŮ

➔ Věděli byste na jakém principu pracuje ultrazvukový senzor?

Základním principem je odraz zvukových vln vycházejících ze senzoru od okolních předmětů a jejich následné detekci. Měří se čas, který uplynul od odeslání k přijmutí.



Žáci ať zapojí ultrazvukový senzor podle zobrazeného schématu, který je součástí pracovních listů, nebo přiložené prezentace, kterou lze promítat pomocí dataprojektoru.

UPOZORNĚNÍ

→ Upozorněte žáky na to, ať se soustředí na správné zapojení pinů ultrazvukového senzoru Trig a Echo.



2. KROK 🕒 15 minut

Nyní studentům ukažte prostřednictvím dataprojektoru nebo pracovního listu základní kód, pro implementaci ultrazvukového senzoru.

```
1  #define trigPin 2
2  #define echoPin 3
3  long duration;
4  int distance;
5
6  void setup() {
7      pinMode(trigPin, OUTPUT);
8      pinMode(echoPin, INPUT);
9      Serial.begin(9600);
10 }
11
12 void loop() {
13     digitalWrite(trigPin, LOW);
14     delayMicroseconds(5);
15     digitalWrite(trigPin, HIGH);
16     delayMicroseconds(10);
17     digitalWrite(trigPin, LOW);
18     duration = pulseIn(echoPin, HIGH);
19     distance = duration * 0.034 / 2;
20     Serial.print("Distance = ");
21     Serial.print(distance);
22     Serial.println(" cm");
23     delay(50);
24 }
```

RYCHLÝ TIP

- Ať žáci vysvětlí implementaci vzorce pro zjištění vzdálenosti předmětu od ultrazvukového čidla.
- $\text{Vzdálenost (cm)} = \text{rychlost zvuku (cm/}\mu\text{s)} \times \text{čas (}\mu\text{s)}/2$



Žáci ať program nahrají do desky a odzkouší, zda se v sériovém monitoru objevuje vzdálenost od překážek.

3. KROK 10 minut

Na základě zvládnutí principů ovládání motoru, budou studenti řešit následující úkol.

ÚKOL PRO STUDENTY

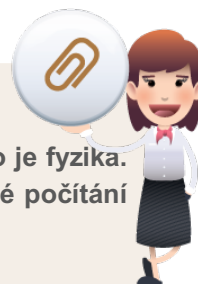
- A) Vyzkoušejte přesnost měření vzdálenosti pomocí ultrazvukového senzoru.

Jedná se o jednoduchý experiment, kdy se žáci vzdalují od senzoru a měří pomocí pravítka reálnou vzdálenost a porovnávají s hodnotou na sériovém monitoru.



RYCHLÝ TIP

- V přechodím úkolu je viditelný přesah i do dalšího předmětu, jako je fyzika. Může se jednat laboratorní práce zaměřené na měření a následné počítání statistický chyb a odchylek.



PRACOVNÍ LIST – ULTRAZVUKOVÝ SENZOR

JEDNÍM Z MNOHA SENZORŮ V ROBOTICE JE ULTRAZVUKOVÉ ČIDLO, KTERÉ NALEZNE UPLATNĚNÍ V CELÉ ŘADĚ APLIKACÍ. JEHO PRINCIP JE POMĚRNĚ JEDNODUCHÝ A TVOŘÍ NEDOMYSLITELNOU ČÁST SENZORICKÉHO SUBSYSTÉMU.

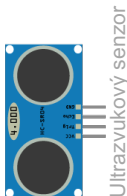
CO SE NAUČÍTE

- ① Pochopení principu ultrazvukového senzoru.
- ② Zapojení ultrazvukového senzoru.
- ③ Programování ultrazvukového senzoru.



CO BUDETE POTŘEBOVAT

- ① Ultrazvukové čidlo
- ② Desku Arduino.
- ③ Kontaktní pole.
- ④ Vodiče typu zástrčka-zástrčka.



POUŽITÉ SOUČÁSTKY

OTÁZKA PRO VÁS

→ Věděli byste na jakém principu pracuje ultrazvukový senzor?

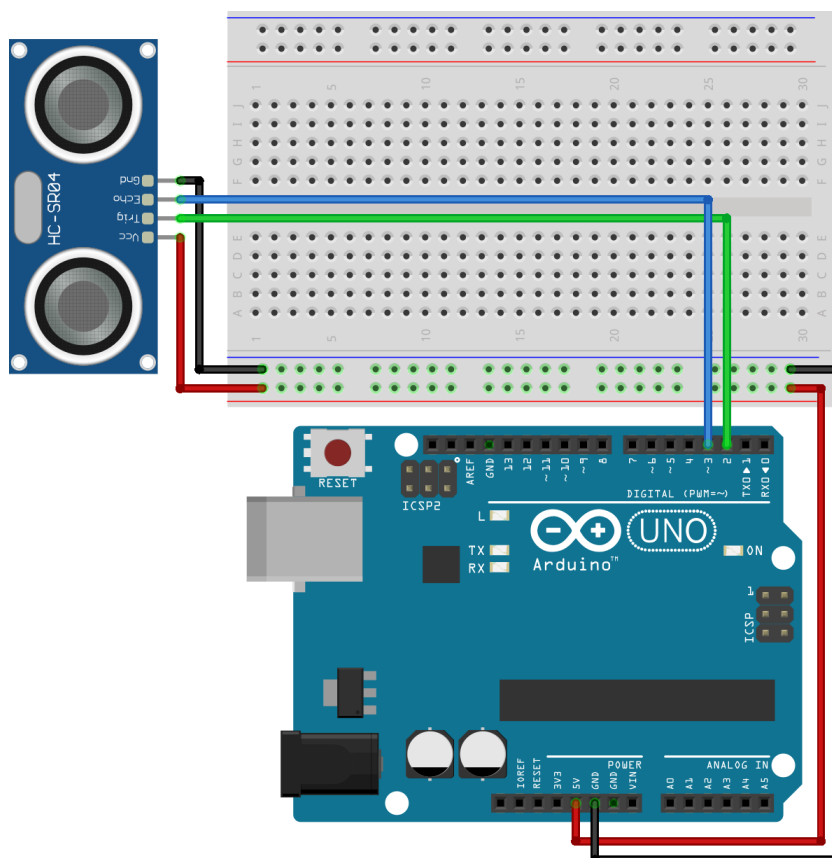


A JDĚTE NA TO ...

- ① Podle přiloženého schématu zapojte obvod ultrazvukovým senzorem.

DEJTE SI POZOR

→ Pozor si dejte zejména na zapojení pinů Trig a Echo.



- ② Napište program, který zprovozní ultrazvukové čidlo, které bude zobrazovat vzdálenost od objektu.

```
1  #define trigPin 2
2  #define echoPin 3
3  long duration;
4  int distance;
5
6  void setup() {
7      pinMode(trigPin, OUTPUT);
8      pinMode(echoPin, INPUT);
9      Serial.begin(9600);
10 }
11
12 void loop() {
13     digitalWrite(trigPin, LOW);
14     delayMicroseconds(5);
15     digitalWrite(trigPin, HIGH);
16     delayMicroseconds(10);
17     digitalWrite(trigPin, LOW);
18     duration = pulseIn(echoPin, HIGH);
19     distance = duration * 0.034 / 2;
20     Serial.print("Distance = ");
21     Serial.print(distance);
22     Serial.println(" cm");
23     delay(50);
24 }
```

OTÁZKA PRO VÁS

➔ Dokázali byste vysvětlit vzorec, který vypočítává vzdálenost čidla od objektu?



- ③ Pokud máte program napsaný, tak jej nahrajte do desky Arduino.
- ④ Spusťte sériový monitor a vyzkoušejte, zda čidlo reaguje na předměty, které umísťujete v různé vzdálenosti.



ÚKOL PRO VÁS

→ Ověřte přesnost ultrazvukového senzoru. Postupně pokládejte libovolný předmět před senzor v různých vzdálenostech. Měřte tuto vzdálenost pravítkem. Naměřené hodnoty porovnávejte se zobrazenými daty na sériovém monitoru. Údaje pro porovnání zadávejte do tabulky.

PRŮVODCE HODINOU II



Žáci budou samostatně pracovat na projektech spojených s používáním ultrazvukového čidla. Využijí k tomu znalost zapojení senzoru a programových konstrukcí z předešlých lekcí.



PŘÍPRAVA

Co bude v této hodině potřeba?

- ① Součásti obvodu – deska Arduino s USB kabelem, kontaktní pole, ultrazvukový senzor, vodiče typu zástrčka-zástrčka, LED, rezistory.
- ② Osobní počítač pro studenty s nainstalovaným Arduino IDE.
- ③ Pokud je k dispozici, tak dataprojektor.
- ④ Prezentace k lekci 11.
- ⑤ Pracovní listy pro studenty.

1. KROK 🕒 10 minut

Na úvod rozdejte studentům sady Arduino. Řekněte, že náplní vašeho kurzu bude procvičování práce s ultrazvukovým senzorem formou praktických příkladů.

ZEPTEJTE SE STUDENTŮ

➔ Jaká je konstrukce podmínkového příkazu `if`?

Jedná se o zopakování struktury podmínkového příkazu `if`, který bude použit v samostatném úkolu.



Žáci ať v rámci opakování zapojí ultrazvukový senzor podle schématu z minulé hodiny. Pro připomenutí lze zobrazit schéma obvodu pomocí dataprojektoru nebo v rámci pracovních listů.

2. KROK 10 minut

Nyní studentům ukažte prostřednictvím dataprojektoru nebo pracovního listu schéma zapojení pro samostatný úkol.

Žáci ať obvod složí.

3. KROK 25 minut

Na základě zvládnutí základní práce s ultrazvukovým senzorem, budou studenti řešit následující úkol.



ÚKOL PRO STUDENTY

- A) Naprogramujte obvod se dvěma diodami tak, aby se červená dioda rozsvítila, pokud bude překročena nastavená minimální vzdálenost např. 10 cm od překážky. Zelená dioda se rozsvítí při naměřené vzdálenosti větší, jak 100 cm.

PRACOVNÍ LIST – ULTRAZVUKOVÝ SENZOR POKRAČOVÁNÍ

JEDNÍM Z MNOHA SENZORŮ V ROBOTICE JE ULTRAZVUKOVÉ ČIDLO, KTERÉ NALEZNE UPLATNĚNÍ V CELÉ ŘADĚ APLIKACÍ. JEHO PRINCIP JE POMĚRNĚ JEDNODUCHÝ A TVOŘÍ NEDOMYSLITELNOU ČÁST SENZORICKÉHO SUBSYSTÉMU.

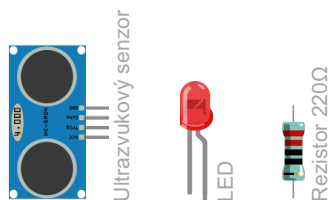
CO SE NAUČÍTE

- ① Propojit základní kód pro ovládání senzoru s podmínkovým příkazem.
- ② Zapojení ultrazvukového senzoru do obvodu s dalšími součástkami.
- ③ Propojení vstupního signálu ultrazvukového senzoru s ovládáním dalších komponent.



CO BUDETE POTŘEBOVAT

- ① Ultrazvukové čidlo.
- ② Desku Arduino.
- ③ Kontaktní pole.
- ④ Červená a zelená LED.
- ⑤ Rezistory 220 Ohm.
- ⑥ Vodiče typu zástrčka-zástrčka.



POUŽITÉ SOUČÁSTKY

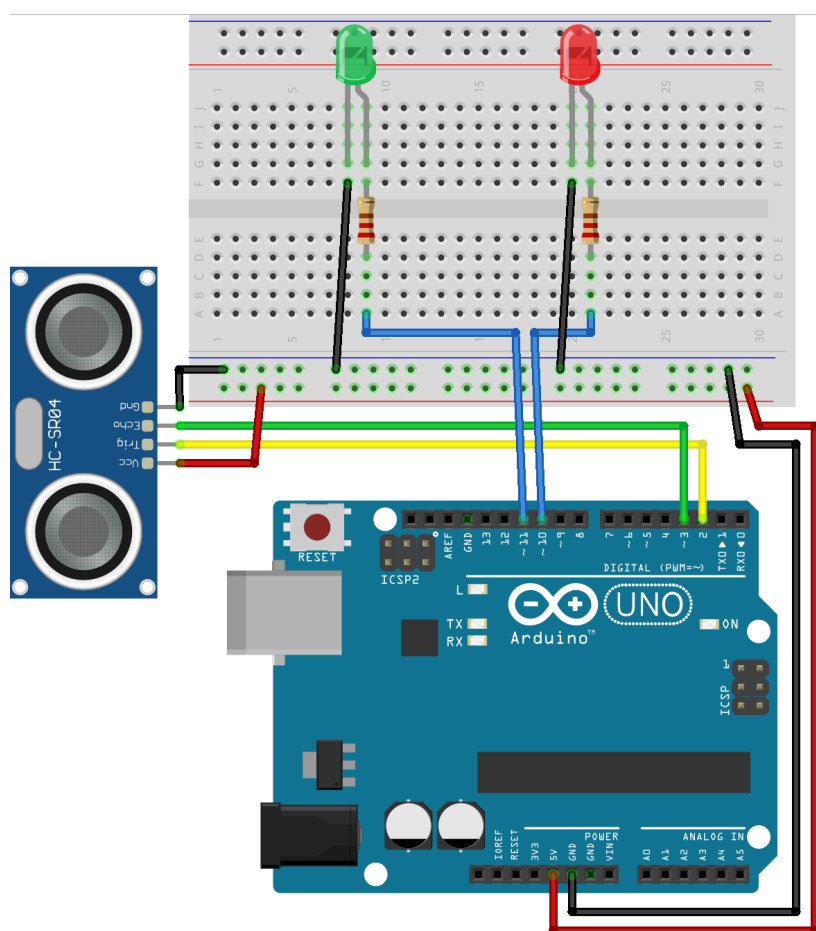
OTÁZKA PRO VÁS

→ Jaká je konstrukce podmínkového příkazu `if`?



A JDĚTE NA TO ...

- 1 Podle přiloženého schématu zapojte obvod s ultrazvukovým senzorem.





ÚKOL PRO VÁS

- A) Naprogramujte obvod se dvěma diodami tak, aby se červená dioda rozsvítila, pokud bude překročena nastavená minimální vzdálenost např. 10 cm od překážky. Zelená dioda se rozsvítí při naměřené vzdálenosti větší, jak 100 cm.

ŘEŠENÍ ÚLOH

Úkol A)

```
1  #define trigPin 2
2  #define echoPin 3
3  #define ledMin 10
4  #define ledMax 11
5  int maximumRange = 200;
6  int minimumRange = 0;
7  long duration, distance;
8
9  void setup(){
10     Serial.begin(9600);
11     pinMode(trigPin, OUTPUT);
12     pinMode(echoPin, INPUT);
13     pinMode(ledMin, OUTPUT);
14     pinMode(ledMax, OUTPUT);
15 }
16 void loop(){
17     digitalWrite(trigPin, LOW);
18     delayMicroseconds(2);
19     digitalWrite(trigPin, HIGH);
20     delayMicroseconds(10);
21     digitalWrite(trigPin, LOW);
22     duration = pulseIn(echoPin, HIGH);
23     distance = duration * 0.034/2;
24     if(distance < 10){
25         digitalWrite(ledMin, HIGH);
26         digitalWrite(ledMax, LOW);
27     }else if(distance > 100){
28         digitalWrite(ledMax, HIGH);
29         digitalWrite(ledMin, LOW);
30     }else{
31         digitalWrite(ledMin, LOW);
32         digitalWrite(ledMax, LOW);
33     }
34     if(distance >= maximumRange || distance <= minimumRange){
35         Serial.println("Mimo dosah");
36     }else{
37         Serial.print(distance);
38         Serial.println(" cm");
39     }
40     delay(500);
41 }
```

PRŮVODCE HODINOU III



Žáci budou samostatně pracovat na projektech spojených s používáním ultrazvukového čidla. Využijí k tomu znalost zapojení senzoru a programových konstrukcí z předešlých lekcí.



PŘÍPRAVA

Co bude v této hodině potřeba?

- ① Součásti obvodu – deska Arduino s USB kabelem, kontaktní pole, ultrazvukový senzor, vodiče typu zástrčka-zástrčka, LCD displej, potenciometr.
- ② Osobní počítač pro studenty s nainstalovaným Arduino IDE.
- ③ Pokud je k dispozici, tak dataprojektor.
- ④ Prezentace k lekci 11.
- ⑤ Pracovní listy pro studenty.

1. KROK 🕒 10 minut

Na úvod rozdejte studentům sady Arduino. Řekněte, že náplní vašeho kurzu bude procvičování práce s ultrazvukovým senzorem formou praktických příkladů.

RYCHLÝ TIP

- ➔ Pro tuto hodinu lze využít předchozího projektu, který se zabýval snímáním teploty a zobrazováním hodnot na LCD displeji.
- ➔ Může se jednat o doplňkovou hodinu pro procvičení zapojení LCD displeje a následné zobrazení hodnot z ultrazvukového senzoru.



Žáci ať v rámci opakování zapojí ultrazvukový senzor podle schématu z minulé hodiny. Pro připomenutí lze zobrazit schéma obvodu pomocí dataprojektoru nebo v rámci pracovních listů.



TIP

Je možné zobrazit celé schéma zapojení včetně zapojení LCD displeje a následně se soustředit pouze na programový kód.

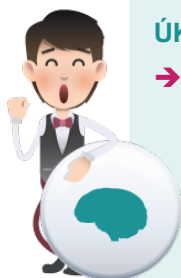
2. KROK 🕒 10 minut

Nyní studentům ukažte prostřednictvím dataprojektoru nebo pracovního listu schéma zapojení pro samostatný úkol.

Žáci ať obvod zapojí.

3. KROK 🕒 25 minut

Na základě zvládnutí základní práce s ultrazvukovým senzorem, budou studenti řešit následující úkol.



ÚKOL PRO STUDENTY

- ➔ A) K ultrazvukovému senzoru připojte LCD displej, na kterém se bude zobrazovat aktuální vzdálenost od překážky. Pro zapojení LCD displeje využijte lekci 6.
- B) Naprogramujte obvod ultrazvukového senzoru a LCD displeje tak, aby se na displeji zobrazoval aktuální údaj o vzdálenosti objektu od senzoru.

PRACOVNÍ LIST – ULTRAZVUKOVÝ SENZOR S LCD DISPLEJEM

JEDNÍM Z MNOHA SENZORŮ V ROBOTICE JE ULTRAZVUKOVÉ ČIDLO, KTERÉ NALEZNE UPLATNĚNÍ V CELÉ ŘADĚ APLIKACÍ. JEHO PRINCIP JE POMĚRNĚ JEDNODUCHÝ A TVOŘÍ NEDOMYSLITELNOU ČÁST SENZORICKÉHO SUBSYSTÉMU.

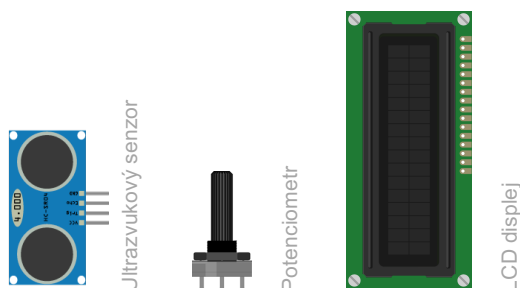
CO SE NAUČÍTE

- ① Zobrazit hodnoty z ultrazvukového senzoru na LCD displeji.
- ② Zopakování práce s LCD displejem.



CO BUDETE POTŘEBOVAT

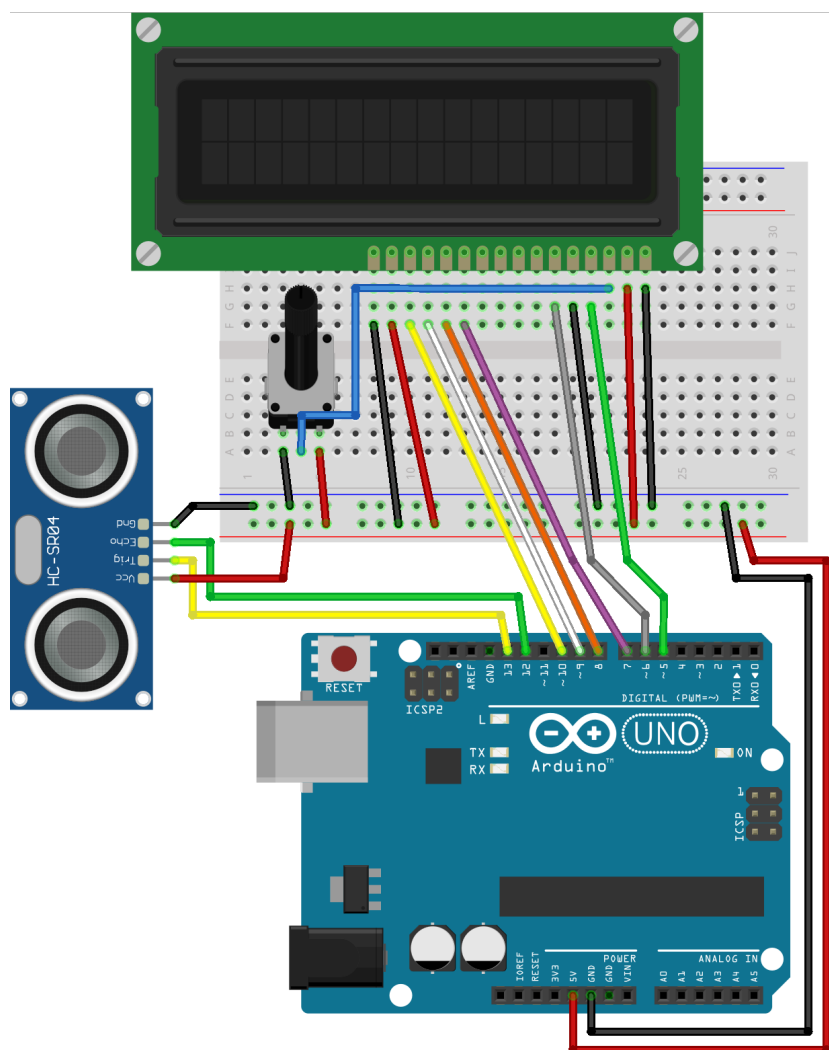
- ① Ultrazvukové čidlo.
- ② Desku Arduino.
- ③ Kontaktní pole.
- ④ LCD displej.
- ⑤ Potenciometr.
- ⑥ Vodiče typu zástrčka-zástrčka.



POUŽITÉ SOUČÁSTKY

A JDĚTE NA TO ...

- ① Podle přiloženého schématu zapojte obvod s ultrazvukovým senzorem a LCD displejem.



ÚKOL PRO VÁS

- B) Naprogramujte obvod ultrazvukového senzoru a LCD displeje tak, aby se na displeji zobrazoval aktuální údaj o vzdálenosti objektu od senzoru.

ŘEŠENÍ ÚLOH

Úkol B)

```
1  #include <LiquidCrystal.h>
2
3  const int trigPin = 3;
4  const int echoPin = 4;
5
6  const int rsPin = 5;
7  const int ePin = 6;
8  const int d4Pin = 7;
9  const int d5Pin = 8;
10 const int d6Pin = 9;
11 const int d7Pin = 10;
12 LiquidCrystal LCD(rsPin,ePin,d4Pin,d5Pin,d6Pin,d7Pin);
13
14 void setup()
15 {
16     pinMode(trigPin, OUTPUT);
17     pinMode(echoPin, INPUT);
18
19     LCD.begin(16,2);
20     LCD.clear();
21     LCD.setCursor(0,0);
22     LCD.print("Vzdálenost: ");
23 }
24
25 void loop()
26 {
27     digitalWrite(trigPin, HIGH);
28     delayMicroseconds(10);
29     digitalWrite(trigPin, LOW);
30     duration = pulseIn(echoPin, HIGH);
31     distance = duration * 0.034 / 2;
32
33     LCD.setCursor(13,0);
34     LCD.print(distance);
35     LCD.print(" cm");
36
37     delay(10);
38 }
```

PRŮVODCE HODINOU IV



Žáci budou samostatně pracovat na závěrečném projektu, ve kterém využijí předchozích znalostí spojených s používáním ultrazvukového čidla a ovládání stejnosměrných motorů. Výsledkem projektu by měl být robotický prostředek v podobě vozítka, které se bude vyhýbat překážkám.



PŘÍPRAVA

Co bude v této hodině potřeba?

- ① Součásti obvodu – deska Arduino s USB kabelem, kontaktní pole, ultrazvukový senzor, vodiče typu zástrčka-zástrčka, stejnosměrné motory, řídicí obvod L293D, kolový podvozek.
- ② Osobní počítač pro studenty s nainstalovaným Arduino IDE.
- ③ Pokud je k dispozici, tak dataprojektor.
- ④ Prezentace k lekci 11.
- ⑤ Pracovní listy pro studenty.



TIP

Pro tento projekt je vhodné, aby si žáci vyrobili svůj kolový podvozek se dvěma motory, na který se bude aplikovat elektronická část. Může se jednat o zcela unikátní, vlastní řešení, pro které mohou využít libovolný materiál.

1. KROK 🕒 5 minut

Na úvod rozdejte studentům sady Arduino. Předpokládáme, že žáci budou mít podvozky. Potom řídicí subsystém mohou sestavit z komponent sady Arduino. Řekněte, že náplní hodiny bude naprogramování jednoduchého, autonomního robotu, který se bude vyhýbat překážkám. senzorem formou praktických příkladů.

RYCHLÝ TIP

➔ Pro tuto hodinu lze využít předchozích projektů, ve kterých se programoval ultrazvukový senzor. Může být použita forma opakování, tzn. Žáci zapojí a naprogramují senzor v rámci opakování, nebo použijí některý z již vytvořených zdrojových souborů.



2. KROK 🕒 10 minut

Nyní žákům ukažte prostřednictvím dataprojektoru nebo pracovního listu schéma zapojení celého řídicího subsystému, který obsahuje ultrazvukový senzor, řídicí obvod pro motory a stejnosměrné motory.

Žáci ať obvod zapojí.

TIP

Pokud by byla časová dotace na lekci větší, tak tento krok může být rozdělen do dvou částí. První by se zabývala v rámci opakování zapojením ultrazvukového senzoru a druhá zapojením a ovládáním stejnosměrných motorů.



OTÁZKA PRO ŽÁKY

➔ Jaký je význam řídicího obvodu L293D?

Jedná se o integrovaný obvod, který umožňuje řídit motory v obou směrech. Jedná se o typický ovladač stejnosměrných motorů.



3. KROK 🕒 30 minut

Hlavním úkolem žáků je vymyslet řešení následujícího úkolu.



ÚKOL PRO STUDENTY

→ A) Naprogramujte robotický prostředek tak, aby se při detekci překážky, která je blíže než 20 cm této překážce vyhnul a pokračoval v libovolném směru.

PRACOVNÍ LIST – AUTONOMNÍ VOZÍTKO

PŘI POUŽÍVÁNÍ ULTRAZVUKOVÉHO SENZORU SE PŘÍMO VYBÍZÍ VYTVOŘIT AUTONOMNÍ VOZÍTKO, KTERÉ SE BUDE UMĚT VYHÝBAT PŘEKÁŽKÁM. A TO JETAKÉ TÉMA TÉTO LEKCE. VŽDYŤ I V REÁLNÍM PROVOZU NYNÍ AUTOMOBILY DISPONUJÍ TÍMTO BEZPEČNOSTNÍM PRVKEM.

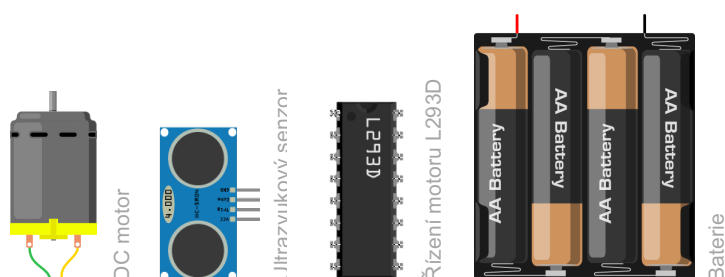
CO SE NAUČÍTE

- 1 Propojit ovládání stejnosměrných motorů s ultrazvukovým čidlem.
- 2 Naprogramovat logiku autonomního vozítka.
- 3 Aplikovat vlastní konstrukci vozítka do reálné podoby.



CO BUDETE POTŘEBOVAT

- 1 Ultrazvukové čidlo.
- 2 Desku Arduino.
- 3 Kontaktní pole.
- 4 Řídící obvod L293D.
- 5 Stejnosměrné motory.
- 6 Vodiče typu zástrčka-zástrčka.
- 7 Externí zdroj napájení – baterie.



POUŽITÉ SOUČÁSTKY

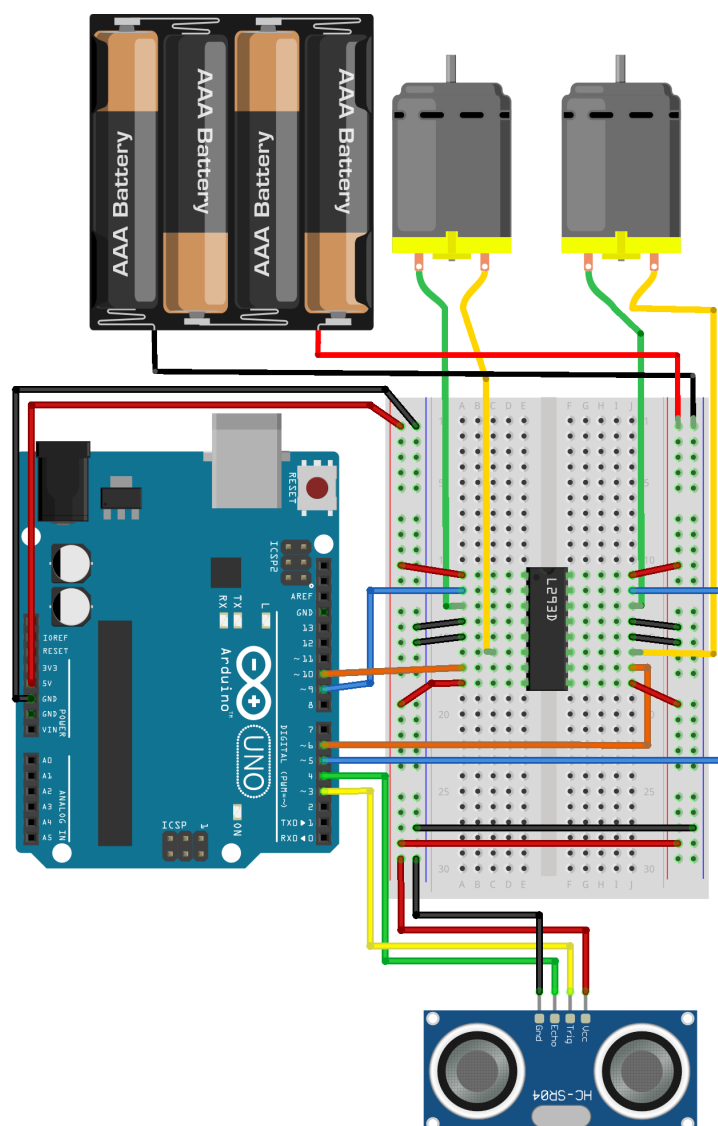


PŘÍPRAVA

Pro realizaci tohoto projektu si připravte robotický podvozek. Ten může být vyroben z libovolných dílů. Může se jednat o komponenty z různých stavebnic, dřevěné prvky, staré CD nosiče jako kola apod.

A JDĚTE NA TO ...

- ① Podle přiloženého schématu zapojte obvod s ultrazvukovým senzorem a dvěma stejnosměrnými motory.





ÚKOL PRO VÁS

→ A) Naprogramujte obvod ultrazvukového senzoru a řízení stejnosměrných motorů tak, aby se robotické vozítko dokázalo vyhnout předmětům a pokračovalo v libovolném směru. Může se jednat o náhodný pohyb po ploše, na které jsou umístěny překážky.

ŘEŠENÍ ÚLOH

Úkol A)

```
1  const int trigPin = 3;
2  const int echoPin = 4;
3  const int leftForward = 9;
4  const int leftBackward = 10;
5  const int rightForward = 5;
6  const int rightBackward = 6;
7
8  int duration = 0;
9  int distance = 0;
10
11 void setup() {
12     pinMode(trigPin, OUTPUT);
13     pinMode(echoPin, INPUT);
14     pinMode(leftForward, OUTPUT);
15     pinMode(leftBackward, OUTPUT);
16     pinMode(rightForward, OUTPUT);
17     pinMode(rightBackward, OUTPUT);
18     Serial.begin(9600);
19 }
20
21 void loop() {
22     digitalWrite(trigPin, HIGH);
23     delayMicroseconds(10);
24     digitalWrite(trigPin, LOW);
25
26     duration = pulseIn(echoPin, HIGH);
27     distance = duration * 0.034 / 2;
28
29     if (distance < 20) {
30         digitalWrite(leftForward, LOW);
31         digitalWrite(leftBackward, HIGH);
32         digitalWrite(rightForward, HIGH);
33         digitalWrite(rightBackward, LOW);
34         delay(100);
35     }else{
36         digitalWrite(leftForward, HIGH);
37         digitalWrite(leftBackward, LOW);
38         digitalWrite(rightForward, HIGH);
39         digitalWrite(rightBackward, LOW);
40     }
41 }
```

PODROBNÝ PRŮVODCE TEORIÍ

UKÁZKA PRÁCE JEDNÍM ZE ZÁKLADNÍCH SENZORŮ, KTERÝ LZE ZAPOJIT S DESKOU ARDUINO. JEDNÁ SE O ULTRAZVUKOVÝ SENZOR VZDÁLENOSTI, KTERÝ POSKYTUJE VÝZNAMNÉ FUNKCIONALITĚ SPOJENÉ S AUTONOMIÍ ROBOTICKÝCH PROSTŘEDKŮ.

OBSAH PRŮVODCE

- ① Pochopení principu ultrazvukového senzoru
- ② Zapojení senzoru vzdálenosti.
- ③ Využití externí knihovny pro jednodušší programování.
- ④ Využití senzoru v různých příkladech.
- ⑤ Projekt robotického vozítka.

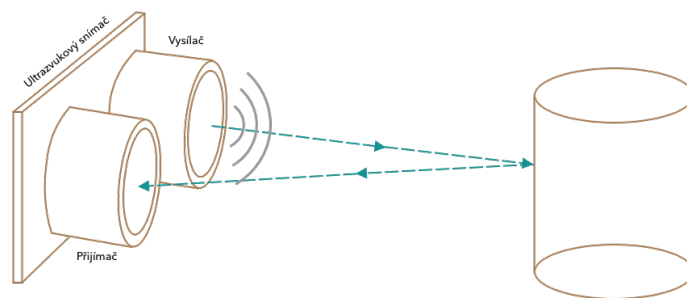
O ULTRAZVUKOVÉM SENZORU

Velmi častým a levným ultrazvukovým senzorem je typ HC-SR04. Jedná se o snadno použitelný ultrazvukový senzor vzdálenosti, s rozsahem od 2 do 400 cm. Běžně se používá při vyhýbání se překážkám robotům a automatizačním projektům.

JAK FUNGUJE ULTRAZVUKOVÝ SENZOR VZDÁLENOSTI?

Ultrazvukové senzory fungují tak, že vydávají zvukové vlny s frekvencí, která je pro člověka příliš vysoká. Tyto zvukové vlny cestují vzduchem rychlostí zvuku, zhruba 343 m / s. Pokud je před snímačem nějaký předmět, zvukové vlny se odrazí zpět a přijímač ultrazvukového snímače je detekuje. Měřením toho, kolik času uplynulo mezi odesláním a příjmem zvukových vln, lze vypočítat vzdálenost mezi senzorem a objektem. Obr. 1 - Princip .

Při 20 °C je rychlost zvuku zhruba 343 m/s nebo 0,034 cm/μs. Řekněme, že doba mezi



Obr. 1 - Princip ultrazvukového senzoru

odesláním a příjmem zvukových vln je 2 000 mikrosekund. Pokud se vynásobí rychlost zvuku časem, kdy zvukové vlny urazily, získá se vzdálenost, kterou zvukové vlny urazily.

$$\text{Vzdálenost} = \text{rychlost} \times \text{čas}$$

Ale to není výsledek, který se hledá. Vzdálenost mezi senzorem a objektem je ve skutečnosti pouze poloviční, protože zvukové vlny putovaly ze senzoru do objektu a zpět z objektu do senzoru. Výsledek se musí vydělit dvěma.

$$\text{Vzdálenost (cm)} = \text{rychlost zvuku (cm/}\mu\text{s)} \times \text{čas (}\mu\text{s)} / 2$$

Příklad:

$$\text{Vzdálenost (cm)} = 0,0343 \text{ (cm/us)} \times 2000 \text{ (us)} / 2 = 34,3 \text{ cm}$$

TEPLOTNÍ ZÁVISLOST RYCHLOSTI ZVUKU

Rychlost zvuku ve skutečnosti silně závisí na teplotě a v mnohem menší míře na vlhkosti vzduchu. Uvádí se, že rychlost zvuku stoupá zhruba o 0,6 m/s na stupeň Celsia. Pro většinu případů při 20 ° C lze použít pouze 343 m / s, ale pokud se má získat přesnější hodnoty, můžete vypočítat rychlost zvuku pomocí následujícího vzorce:

$$V \text{ (m / s)} = 331,3 + (0,606 \times T)$$

V = rychlost zvuku (m / s), T = teplota vzduchu (° C)

Tento vzorec nezahrnuje vlhkost, protože jeho vliv na rychlost zvuku je jen velmi malý.

JAK FUNGUJE HC-SR04

Na přední straně senzoru HC-SR04 jsou dva stříbrné válce (ultrazvukové měniče), jeden je vysílačem zvukových vln a druhý přijímačem. Chceme-li, aby senzor generoval zvukový záblesk, musíme nastavit **Trig pin** na minimálně **10 µs**. Senzor poté vytvoří 8cyklový výbuch ultrazvuku při 40 kHz.

Tento zvukový záblesk se pohybuje rychlostí zvuku, odrazí se zpět a je přijímán přijímačem snímače. Pin **Echo** pak vydává čas, který zvukové vlny prošly v mikrosekundách.

Funkcí **pulseIn()** v kódu Arduino se může použít ke čtení délky pulzu z pinu **Echo**. Poté můžeme pomocí vzorce uvedeného výše vypočítat vzdálenost mezi senzorem a objektem.

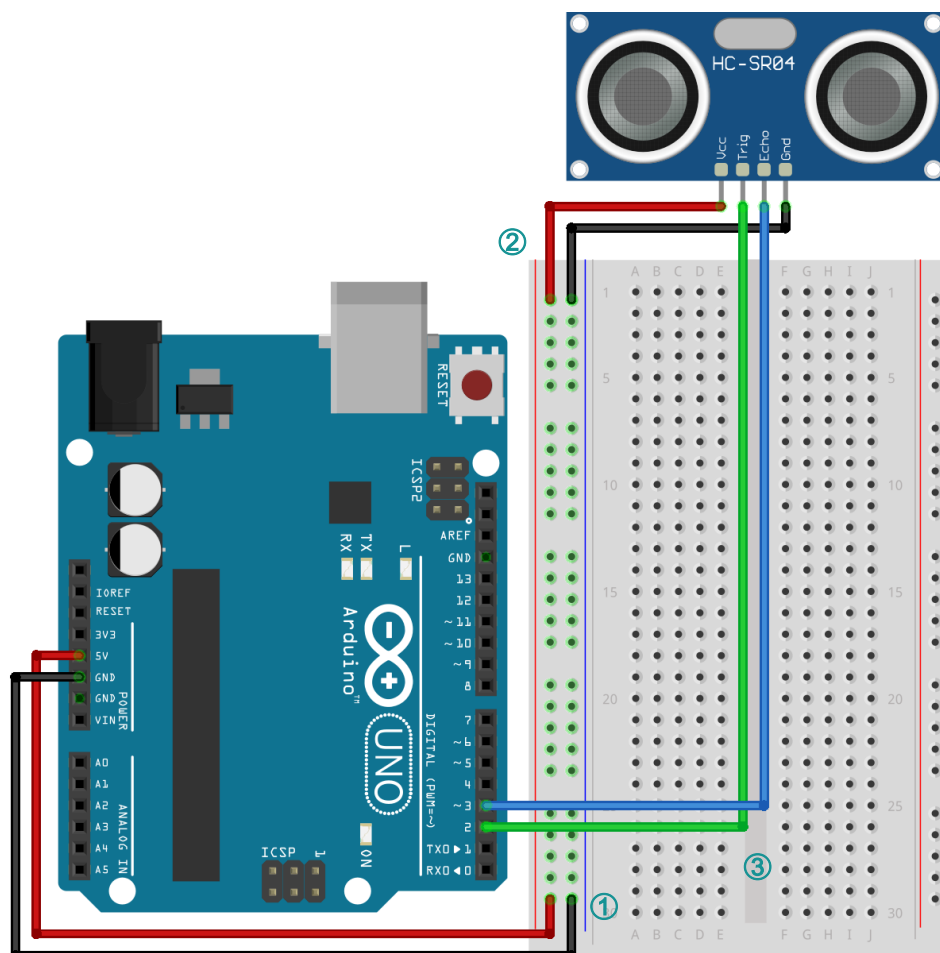


Obr. 2 – Ultrazvukový senzor HC-SR04

ZAPOJENÍ OBVODU

ZÁKLADNÍ PŘÍKLAD

Základní příklad představuje zapojení ultrazvukového senzoru pro jednoduchou ukázkou odečítání vzdálenosti od překážek.



Obr. 3 - Zapojení ultrazvukového senzoru

- ① Do kontaktního pole, do druhé části, přivedeme zem z desky Arduino. Zem z externího napájecího zdroje a z desky Arduino jsou propojeny.
- ② Ze senzoru připojíme pin **VCC** na kladnou polaritu napájení na kontaktní desce a **GND** na zápornou polaritu.
- ③ Pin **ECHO** ze senzoru přivedeme na **pin 2** desky Arduino.
- ④ Pin **TRIG** ze senzoru přivedeme na **pin 3** desky Arduino.

PROGRAMOVÝ KÓD

```
1  #define trigPin 2
2  #define echoPin 3
3
4  long duration;
5  int distance;
6
7  void setup() {
8      pinMode(trigPin, OUTPUT);
9      pinMode(echoPin, INPUT);
10     Serial.begin(9600);
11 }
12
13 void loop() {
14     digitalWrite(trigPin, LOW);
15     delayMicroseconds(5);
16     digitalWrite(trigPin, HIGH);
17     delayMicroseconds(10);
18     digitalWrite(trigPin, LOW);
19     duration = pulseIn(echoPin, HIGH);
20     distance = duration * 0.034 / 2;
21     Serial.print("Distance = ");
22     Serial.print(distance);
23     Serial.println(" cm");
24     delay(50);
25 }
```

- ① Příkazy **#define** definuje konstantní hodnoty. Když je program kompilován, kompilátor nahradí všechny odkazy na tuto konstantu definovanou hodnotou. Takže kdekoli zmíníte **trigPin**, kompilátor jej při kompilaci programu nahradí hodnotou **2**. Stejně tomu je u **echoPin**, kde bude nastavena hodnota **3**.
- ② Dále jsem definovány dvě proměnné: **duration** což je doba trvání mezi odesláním a příjmem zvukových vln. Proměnná **distance** se používá k uložení vypočítané vzdálenosti.
- ③ Ve funkci **setup()** jsou nastaveny konstanty **trigPin** jako výstupu a **echoPin** jako vstup.
- ④ Inicializace sériové komunikace s přenosovou rychlostí 9600. Později lze zobrazit měřenou vzdálenost na sériovém monitoru, který se spustí pomocí **Ctrl+Shift+M** nebo

v nabídce **Nástroje > Sériový monitor**. Zkontrolujte, zda je v sériovém monitoru nastavena přenosová rychlost také na 9600.

- ⑤ Pro získání čistého signálu se začíná vymazáním **trigPinu** nastavením na hodnotu **LOW** na 5 mikrosekund.
- ⑥ Ve funkci **loop()** se spustí senzor nastavením **trigPin** na hodnotu **HIGH** na 10 μ s.
- ⑦ Dále musíme přechíst délku pulzu odeslaného v **echoPin**. K tomu se používá funkce **pulseIn()**. Tato funkce čeká, až se pin přepne z **LOW** na **HIGH**, spustí časování, a poté čeká, až pin přejde na **LOW** a zastaví časování. Poté lze vzdálenost vypočítat pomocí vzorce uvedeného v úvodu.
- ⑧ Nakonec se vypočítaná vzdálenost vytiskne na sériovém monitoru.



Nezapomeňte program zkompileovat a nahrát do desky Arduino. Pokud je vše v pořádku, na sériovém monitoru, který spustíte pomocí **Ctrl+Shift+M** nebo v nabídce **Nástroje > Sériový monitor**. Zkontrolujte, zda je v sériovém monitoru nastavena přenosová rychlost také na **9600**.



NA SÉRIOVÉM MONITORU SE NEZOBRAZUJÍ ŽÁDNÝ TEXT

Nastavení přenosové rychlosti – zkontrolujte, zda je přenosová rychlost nastavena na 9600.

Zapojení senzoru – zkontrolujte zapojení pinu **ECHO** a **TRIG** na senzoru a desce Arduino.

NEJDE NAHRÁT KÓD DO DESKY

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

Správný port – ujistěte se, že máte vybraný správný port pro připojení k desce Arduino pomocí USB kabelu.

ZÁKLADNÍ PŘÍKLAD – VYUŽITÍ KNIHOVNY NEWPING

Knihovnu **NewPing** lze použít s celou řadou ultrazvukových senzorů vzdálenosti. Můžete si všimnout, že níže uvedený kód, který používá knihovnu **NewPing**, je mnohem kratší než kód, který jsme použili dříve. Kromě toho, knihovna **NewPing** obsahuje některé další funkce. Umožňuje například nastavit maximální vzdálenost ke čtení a má integrovaný mediánový filtr, který výrazně zlepšit přesnost naměřených hodnot senzoru HC-SR04.

INSTALACE KNIHOVNY NEWPING

- ① Stáhněte si knihovnu z repozitáře.
- ② V nabídce IDE prostředí **Arduino Projekt > Přidat knihovnu** zvolte **Přidat ZIP knihovnu ...**
- ③ Z disku vyberte stažený soubor knihovny. Knihovna se automaticky přidá do IDE Arduino.
- ④ Zda se knihovna přidala ověříte pohledem do nabídky **Soubor > Příklady > NewPing**.



Knihovna obsahuje příklady, které lze použít, ale musíme je upravit tak, aby odpovídaly našemu nastavení hardwaru.

PROGRAMOVÝ KÓD

```
1  #include <NewPing.h>                                     ①
2
3  #define trigPin  2                                       ②
4  #define echoPin  3                                       ③
5  #define MAX_DISTANCE 350                                 ④
6  NewPing sonar(trigPin, echoPin, MAX_DISTANCE);
7  float duration, distance;
8  void setup() {
9      Serial.begin(9600);
10 }
11
12 void loop() {
13     delay(50);                                           ⑤
14
15     duration = sonar.ping();
16     distance = (duration / 2) * 0.0343;
17
18     Serial.print("Distance = ");
19     Serial.print(distance);                               ⑥
20     Serial.println(" cm");
21 }
```

- ① Připojení knihovny **NewPing**.
- ② Příkazy **#define** definuje konstantní hodnoty. Když je program kompilován, kompilátor nahradí všechny odkazy na tuto konstantu definovanou hodnotou. Takže kdekoli zmíníte **trigPin**, kompilátor jej při kompilaci programu nahradí hodnotou **2**. Stejně tomu je u **echoPin**, kde bude nastavena hodnota **3**.
- ③ Maximální vzdálenost, na kterou chceme provést impuls (v centimetrech). Maximální vzdálenost snímače je dimenzována na 400-500 cm.
- ④ Nastavení pinů a maximální vzdálenosti pro **NewPing**.
- ⑤ Počkejte 50 ms mezi impulzy (asi 20 impulzů/s). Nejkratší zpoždění mezi impulzy by mělo být 29 ms.
- ⑥ Pokud bude limit snímané vzdálenosti mimo nastavené maximum, bude hodnota **distance=0**.



Můžeme také použít **distance=sonar.ping_cm()** nebo **distance=sonar.ping_in()**, který vrátí naměřenou vzdálenost v celých centimetrech nebo palcích. S touto funkcí se nemusí provádět měření délky a počítat vzdálenost.

JAK POUŽÍVAT DIGITÁLNÍ FILTR Z KNIHOVNY NEWPING

Velmi užitečná věc, kterou knihovna **NewPing** nabízí, je zabudovaný mediánový filtr. Tento filtr může výrazně zlepšit přesnost naměřených hodnot HC-SR04. Funkce **ping_median()** pracuje s několika měřeními v řadě, které.

PROGRAMOVÝ KÓD

```
1  #include <NewPing.h>
2
3  #define trigPin 2
4  #define echoPin 3
5  #define MAX_DISTANCE 350
6  NewPing sonar(trigPin, echoPin, MAX_DISTANCE);
7  float duration, distance;
8  void setup() {
9      Serial.begin(9600);
10 }
11
12 void loop() {
13     delay(50);
14
15     int iterations = 5;
16     duration = sonar.ping_median(iterations);
17     distance = duration * 0.0343 / 2;
18
19     Serial.print("Distance = ");
20     Serial.print(distance);
21     Serial.println(" cm");
}
```

①
②

- ① Nastavení počtu měření.
- ② Volání funkce **ping_median()**.



(Př. 1) Do obvodu s ultrazvukovým senzorem zapojte dvě LED (červenou, zelenou). Červená dioda se rozsvítí, pokud bude překročena nastavená minimální vzdálenost např. 10 cm od překážky. Zelená dioda se rozsvítí při naměřené vzdálenosti větší jak 100 cm.



(Př. 2) K ultrazvukovému senzoru připojte LCD displej, na kterém se bude zobrazovat aktuální vzdálenost od překážky. Pro zapojení LCD displeje využijte kapitulu 6.

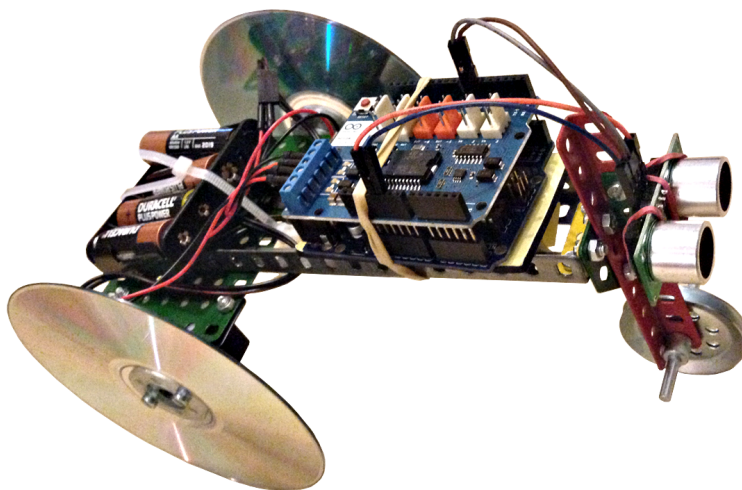
PROJEKT – ROBOTICKÉ VOZÍTKO

Pro vytvoření jednoduchého vozítka můžete využít vše co doma naleznete. Základem jsou dva stejnosměrné motory, které budou tvořit pohonný subsystém. Celková konstrukce může být tvořena například částmi různých stavebnic nebo může být složena z dřevěných prvků. V této části by měli žáci mít zcela volnou ruku. Je to ukázka STEM.

Základní komponenty: 2 stejnosměrné motory, ultrazvukový senzor, modul pro řízení motorů (L293D, nebo motor shield), baterie pro externí napájení, nepájivé pole a vodiče.

KONSTRUKCE

Ukázková konstrukce je složena z komponent stavebnice MERKUR. Jako kola jsou použity staré CD nosiče Obr. 4 – .

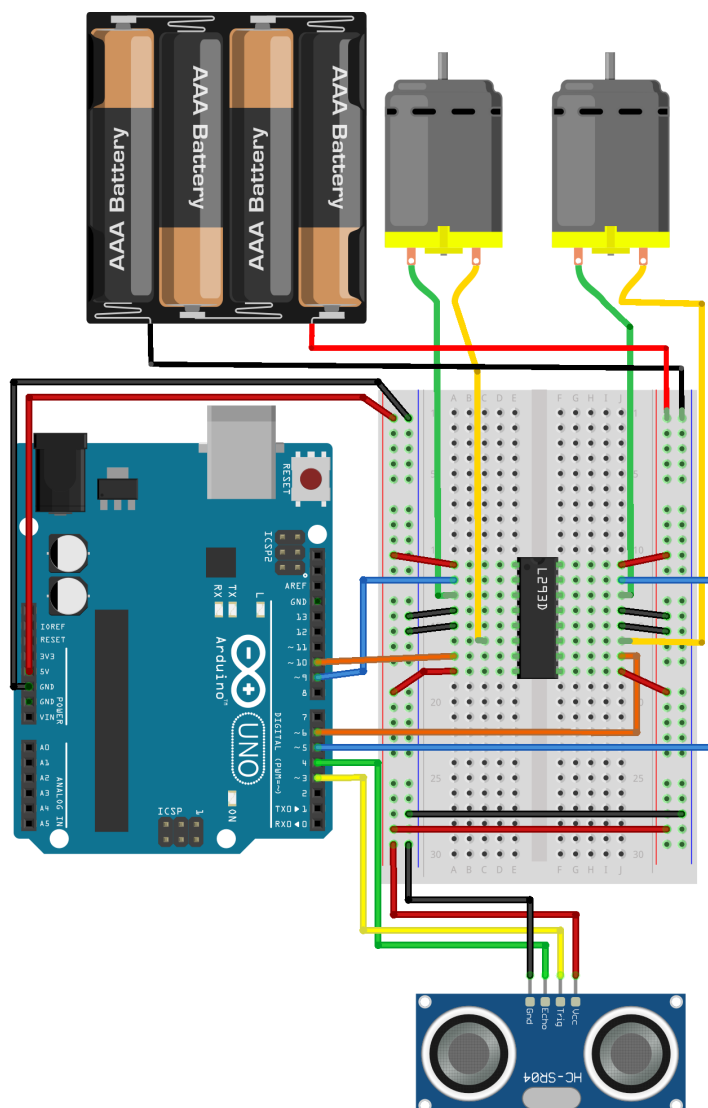


Obr. 4 – Robotický podvozek

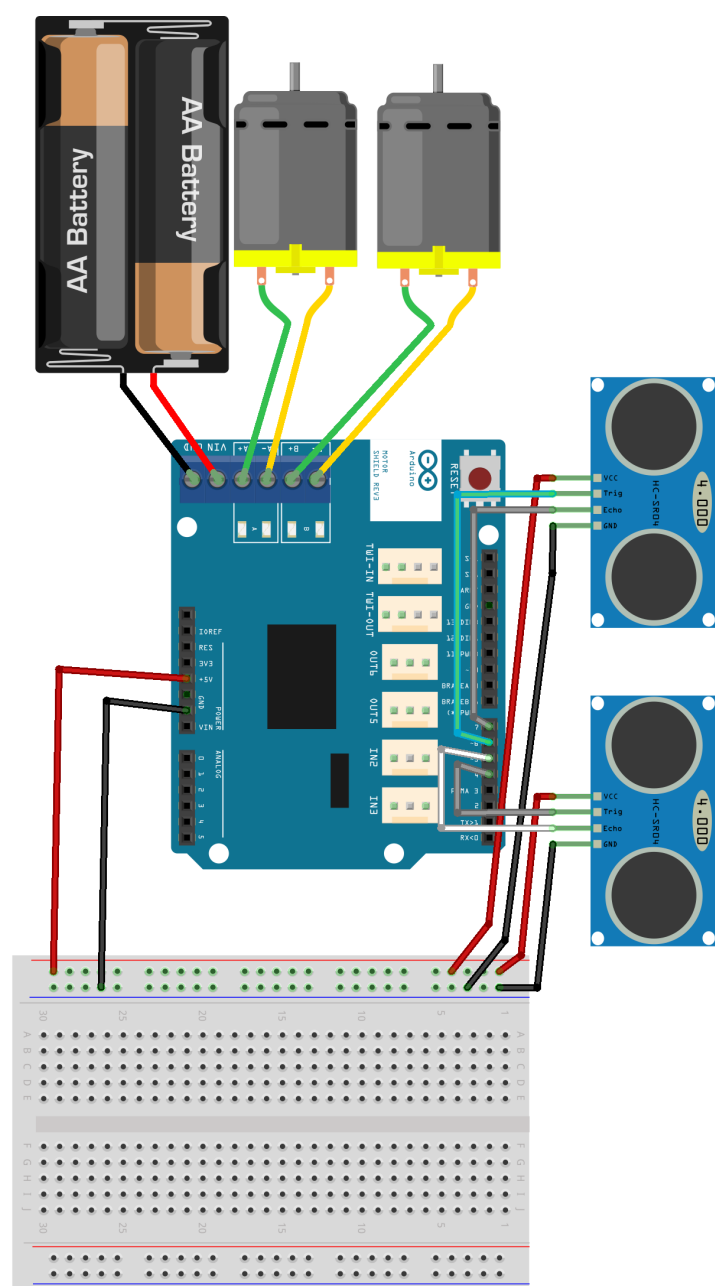
Robotický prostředek by se měl vyhýbat překážkám, které jsou detekovány v určité vzdálenosti pomocí ultrazvukového senzoru. Vyhýbání bude realizováno protisměrným otáčením motorů. Při využití jediného ultrazvukového senzoru se bude robotický prostředek uhýbat vždy na jednu stranu. Vylepšení pak spočívá ve využití dvou těchto senzorů, kdy lze zohlednit i umístění překážky a následně se rozhodnout v jakém směru se prostředek překážce vyhne.

SCHÉMA ZAPOJENÍ

Níže uvedené schéma využívá pro řízení motorů obvodu L293D. Zapojení je tím trochu složitější, ale finančně méně náročné Obr. 5. Při použití pole pro řízení motorů, které se přidává na desku Arduino formou přídatného modulu, se zapojení celého obvodu výrazně zjednoduší Obr. 6.



Obr. 5 – Schéma zapojení



Obr. 6 – Schéma zapojení s motorovým polem a dvěma senzory

PROGRAMOVÝ KÓD

V programovém kódu jsou použity pouze ty konstrukce, které jsou dostupné v předchozích kapitolách.

```
1  const int trigPin = 3;
2  const int echoPin = 4;
3  const int leftForward = 9;
4  const int leftBackward = 10;
5  const int rightForward = 5;
6  const int rightBackward = 6;
7
8  int duration = 0;
9  int distance = 0;
10
11 void setup() {
12     pinMode(trigPin, OUTPUT);
13     pinMode(echoPin, INPUT);
14     pinMode(leftForward, OUTPUT);
15     pinMode(leftBackward, OUTPUT);
16     pinMode(rightForward, OUTPUT);
17     pinMode(rightBackward, OUTPUT);
18     Serial.begin(9600);
19 }
20
21 void loop() {
22     digitalWrite(trigPin, HIGH);
23     delayMicroseconds(10);
24     digitalWrite(trigPin, LOW);
25
26     duration = pulseIn(echoPin, HIGH);
27     distance = duration * 0.034 / 2;
28
29     if (distance < 20) {
30         digitalWrite(leftForward, LOW);
31         digitalWrite(leftBackward, HIGH);
32         digitalWrite(rightForward, HIGH);
33         digitalWrite(rightBackward, LOW);
34         delay(100);
35     }else{
36         digitalWrite(leftForward, HIGH);
37         digitalWrite(leftBackward, LOW);
38         digitalWrite(rightForward, HIGH);
42        digitalWrite(rightBackward, LOW);
43     }
44 }
45
```

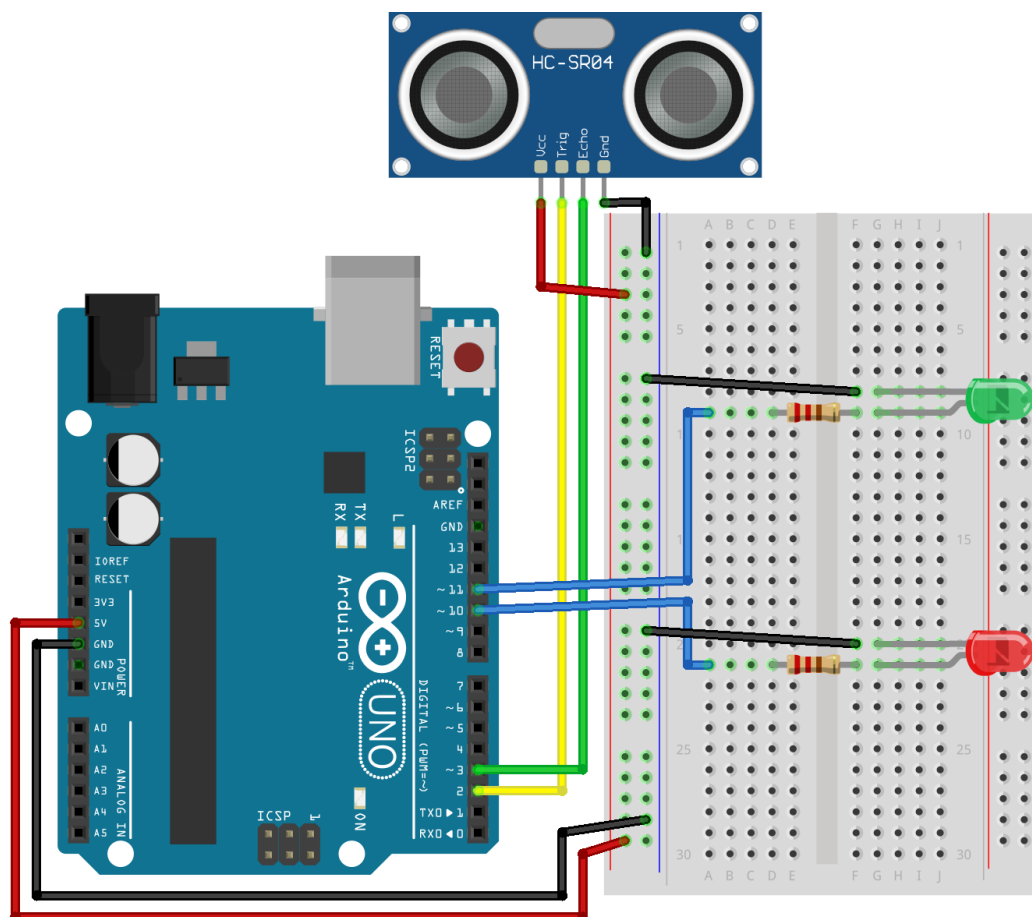
①
②
③
④
⑤
⑥
⑦
⑧

- ① Definice pinů ultrazvukového senzoru.
- ② Definice pinů pro levý motor. Vždy se definují piny pro oba směry – dopřední a zpětný.
- ③ Definice pinů pro pravý motor.
- ④ Nastavení všech použitých pinů jako vstupní **INPUT** a výstupní **OUTPUT**.
- ⑤ Inicializace ultrazvukového senzoru.
- ⑥ Výpočet aktuální vzdálenosti předmětu před ultrazvukovým čidlem.
- ⑦ Na základě definované podmínky dochází ke změně směru otáčení kol. Pokud bude vzdálenost od překážky menší než vzdálenost uvedená v podmínce **if**, nastaví se hodnota pinů tak, aby se motory otáčely proti sobě. Tím dochází k otáčení robotického prostředku.
- ⑧ V opačném případě se robotický prostředek pohybu přímo v dopředném směru.

ŘEŠENÍ PŘÍKLADŮ

PŘÍKLAD 1

Obvod byl upraven tak, že byly přidány dvě diody a rezistory. Tento příklad může být mezistupněm před závěrečným projektem, protože si zde žáci procvičí podmínkový příkaz **if** a práci se vstupy.



Obr. 7 - Zapojení motoru s fotorezistorem

Programový kód využívá základního příkladu a může být pozměněn tak, že se využije externí knihovna **NewPing**. To může být dalším samostatným příkladem.

```

1  #define trigPin 2
2  #define echoPin 3
3  #define ledMin 10
4  #define ledMax 11
5  int maximumRange = 200;
6  int minimumRange = 0;
7  long duration, distance;
8
9  void setup(){
10     Serial.begin (9600);
11     pinMode(trigPin, OUTPUT);
12     pinMode(echoPin, INPUT);
13     pinMode(ledMin, OUTPUT);
14     pinMode(ledMax, OUTPUT);
15 }
16
17 void loop(){
18     digitalWrite(trigPin, LOW);
19     delayMicroseconds(2);
20     digitalWrite(trigPin, HIGH);
21     delayMicroseconds(10);
21     digitalWrite(trigPin, LOW);
23     duration = pulseIn(echoPin, HIGH);
24     distance = duration * 0.034/2;
25
26     if(distance < 10){
27         digitalWrite(ledMin,HIGH);
28         digitalWrite(ledMax,LOW);
29     }else if(distance > 100){
30         digitalWrite(ledMax,HIGH);
31         digitalWrite(ledMin,LOW);
32     }else{
33         digitalWrite(ledMin,LOW);
34         digitalWrite(ledMax,LOW);
35     }
36     if(distance>=maximumRange || distance<=minimumRange){
37         Serial.println("Mimo dosah");
38     }else{
42         Serial.print(distance);
43         Serial.println(" cm");
44     }
45
46     delay(500);
47 }

```

①

②

③

④

⑤

⑥

⑦

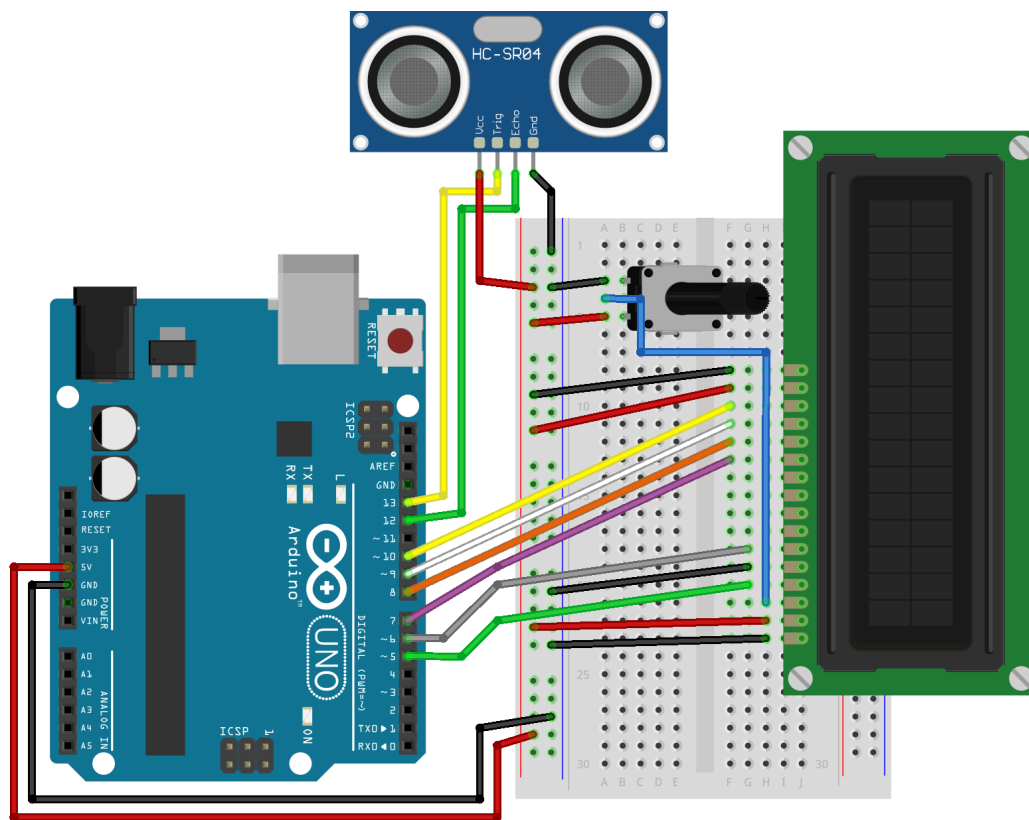
⑧

⑨

- ① Definice pinů pro ultrazvukový senzor.
- ② Definice pinů pro LED. Pin 10 je určen pro červenou a pin 11 pro zelenou diodu.
- ③ Definice proměnných, které vymezují minimální a maximální okruh snímání.
- ④ Přiřazení typu pinů pro ultrazvukový senzor a LED.
- ⑤ Zjištění dat pro výpočet vzdálenosti a samotný výpočet vzdálenosti překážky od ultrazvukového senzoru.
- ⑥ První část podmínky. Jestliže bude vzdálenost předmětu od překážky menší jak 10 cm, potom se rozsvítí červená dioda a zelená bude zhasnutá.
- ⑦ Pokud bude vzdálenost od překážky větší jak 100 cm, potom se rozsvítí zelená dioda a červená zhasne.
- ⑧ Zde se jedná o normální provoz, kdy se překážka pohybuje mezi stanovenými limity, tj. mezi 10 a 100 cm.
- ⑨ Pouze informativní věc, pokud se pohybuje překážka mimo stanovený rozsah.

PŘÍKLAD 2

Pro řešení tohoto příkladu lze vycházet z kapitoly zabývající se zobrazováním teploty a vlhkosti na LCD displeji – lekce 6. Uvedená čidla budou nahrazena ultrazvukovým a na LCD displeji se bude zobrazovat vzdálenost od překážky.



Obr. 8 - Zapojení čidla a LCD displeje

Programový kód opět využívá základního příkladu a může být pozměněn tak, že se využije externí knihovna **NewPing**. To může být dalším samostatným příkladem.

```
1
2  #include <LiquidCrystal.h>
3
4  const int trigPin = 3;
5  const int echoPin = 4;
6
7  const int rsPin = 5;
8  const int ePin = 6;
9  const int d4Pin = 7;
10 const int d5Pin = 8;
11 const int d6Pin = 9;
12 const int d7Pin = 10;
13 LiquidCrystal LCD(rsPin,ePin,d4Pin,d5Pin,d6Pin,d7Pin);
14
15 void setup()
16 {
17     pinMode(trigPin, OUTPUT);
18     pinMode(echoPin, INPUT);
19
20     LCD.begin(16,2);
21     LCD.clear();
22     LCD.setCursor(0,0);
23     LCD.print("Vzdálenost: ");
24 }
25
26 void loop()
27 {
28     digitalWrite(trigPin, HIGH);
29     delayMicroseconds(10);
30     digitalWrite(trigPin, LOW);
31     duration = pulseIn(echoPin, HIGH);
32     distance = duration * 0.034 / 2;
33
34     LCD.setCursor(13,0);
35     LCD.print(distance);
36     LCD.print(" cm");
37
38     delay(10);
39 }
```

①

②

③

④

⑤

⑥

- ① Připojení knihovny pro práci s LCD displejem.
- ② Definice pinů pro LCD displej a inicializace třídy LCD.
- ③ Nastavení pinů pro ultrazvukový senzor.
- ④ Prvotní inicializace LCD displeje.
- ⑤ Aktivace ultrazvukového čidla a výpočet vzdálenosti.
- ⑥ Nastavení kurzoru na LCD displeji pro výpis aktuální hodnoty vzdálenosti.