Kapitola 4 Projekty rozšíření

Tato kapitola je pokročilá a zaměřuje se na hlubokou integraci konstrukčních a řídicích systémů. Jejím cílem je rozvíjet inženýrské konstrukční myšlení a inovační dovednosti studentů. Stavbou inteligentních zařízení s mechanickými strukturami a interaktivní logikou si studenti prostřednictvím praktických cvičení zlepší dovednosti v oblasti systémové integrace a implementace projektů.

Poznámka: Obsah této kapitoly vyžaduje pro učení použití dřevěné tabule QE201. Sadu QE201 si prosím zakupte samostatně.

Lekce 34 Chytrý bionický motýl

🎯 Cíle kurzu

- Pochopte koncept „bioniky“
- Naučte se sestavit chytrý bionický model motýla

-Napište program, který simuluje mávání křídel bionického motýla.

🔍 Úvod do kurzu

V přírodě motýli tančí nejen s elegancí, ale také s úžasnou energetickou účinností. Spoléhají se na svá lehká křídla k provádění složitých manévrů, jako je let, vznášení se a řízení. Vědci se těmito tvory inspirovali k návrhu různých bionických robotů.

V této lekci budeme simulovat třepotavé pohyby motýla a postavíme si vlastního „chytrého bionického motýla“.

Body znalostí

1. Biomimikry

Biomimikry jsou interdisciplinární disciplíny, které studují strukturu, funkci a pohyb živých organismů v přírodě a aplikují své principy na inženýrství, design a vývoj technologií. Pomáhají lidem řešit složité technické problémy napodobováním moudrosti přírody a představují dokonalou kombinaci „biologie“ a „inženýrství“.

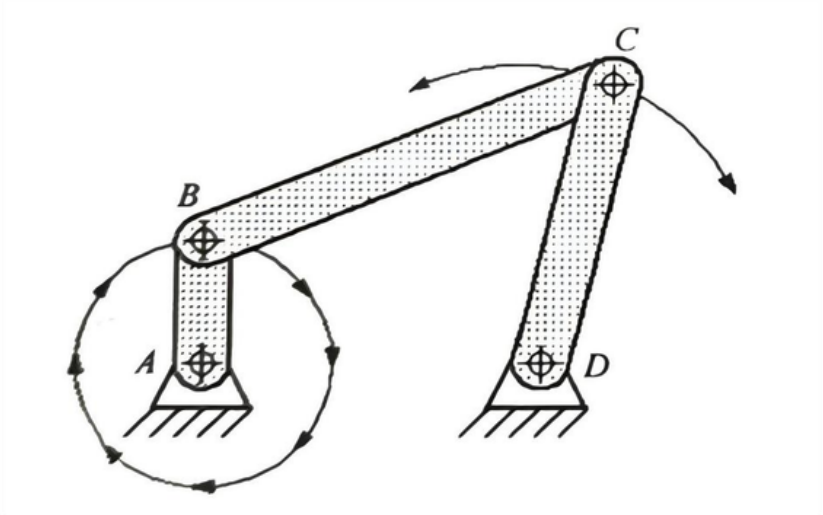
Ve světě biomimikry vědci často pozorují tvory v přírodě, aby se inspirovali pro technologické inovace. Například vážka má při letu vynikající stabilitu a flexibilitu, což inspirovalo design kvadrokoptérových dronů; dlouhý tvar zobáku ledňáčka může snížit odpor při vstupu do vody a používá se v aerodynamickém designu hlavy rychlovlaku. Pevnost pavoučího hedvábí daleko převyšuje pevnost oceli, což vedlo vědce k vývoji vysoce pevných syntetických vláken; chodidla gekonů se mohou volně pohybovat po hladkých stěnách a jejich struktura je napodobována pro výrobu nanoměřítkových biomimetických adhezivních materiálů; lehký a rytmický pohyb křídel motýlů byl použit k vývoji bionických létajících robotů a dekorativního umění s estetickými funkcemi.

2. Inteligentní bionický motýl

Inteligentní bionický motýl kombinuje převážně strukturu převodového mechanismu s klikovým mechanismem, čímž dosahuje bionického efektu mávání křídel a simuluje letové pohyby skutečných motýlů.

Strukturální princip inteligentního bionického motýla

Inteligentní bionický motýl kombinuje převodovou konstrukci s vačkovým mechanismem, čímž dosahuje bionického efektu mávání křídel a simuluje letové pohyby skutečných motýlů.



V ten inteligentní bionický motýl struktura, když ten motor začíná, ten motorotáčí senavysílat moc naten ostatní dva velký ozubená kola podle výstroj zasnoubení. Když tenspojovací tyč je přinesl níže podle A velký výstroj (klika), ten motýlí křídla klapka dolů. Když ten spojovací tyč je přinesl nahorupodle ten velký výstroj, ten motýlí křídla klapka nahoru.

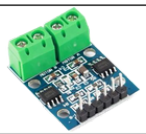
Experiment

1. Popis projektu

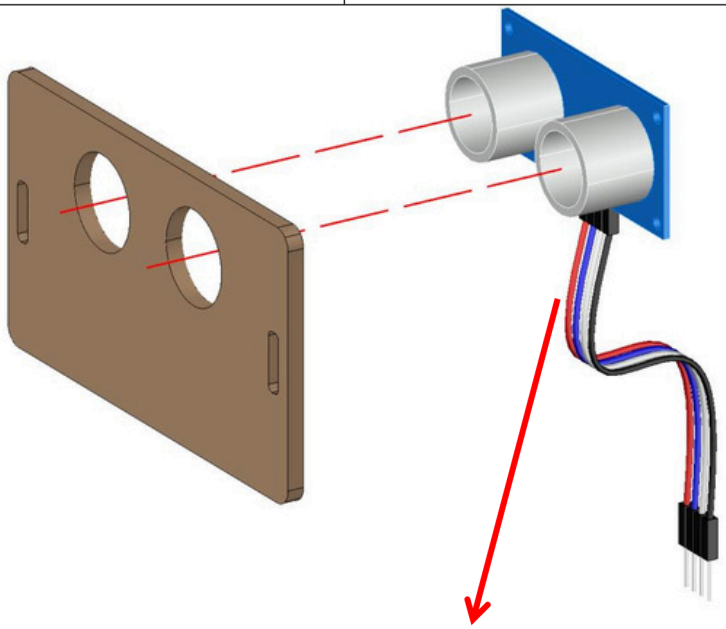
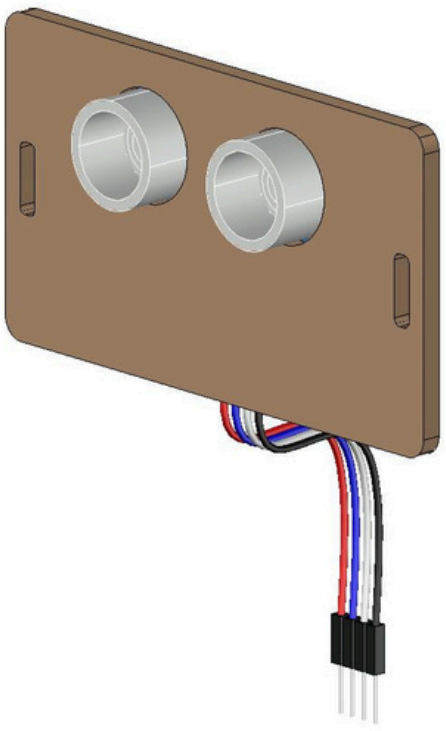
Ultrazvukový modul provádí detekci vzdálenosti v reálném čase. Pokud je vzdálenost objektu před motýlem menší než 20 cm, motýl začne mávat křídly, a pokud je detekční vzdálenost větší než 20 cm, křídla přestanou mávat.

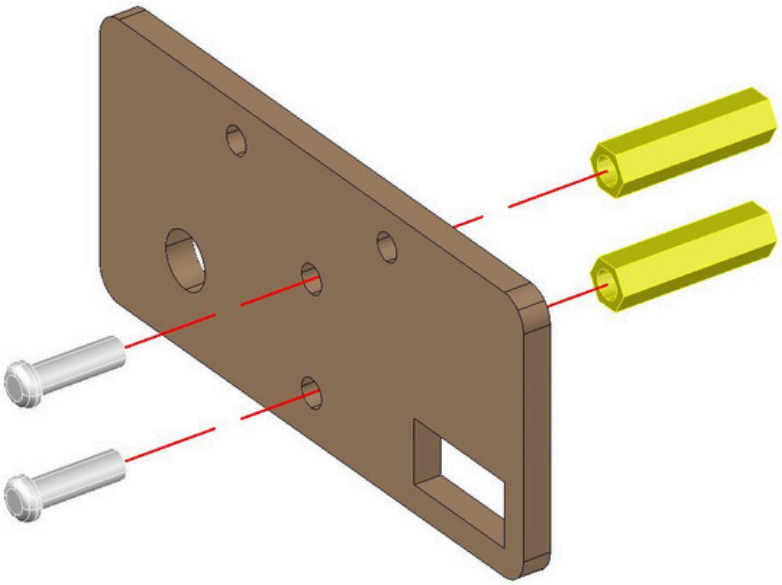
2. Seznam hardwaru

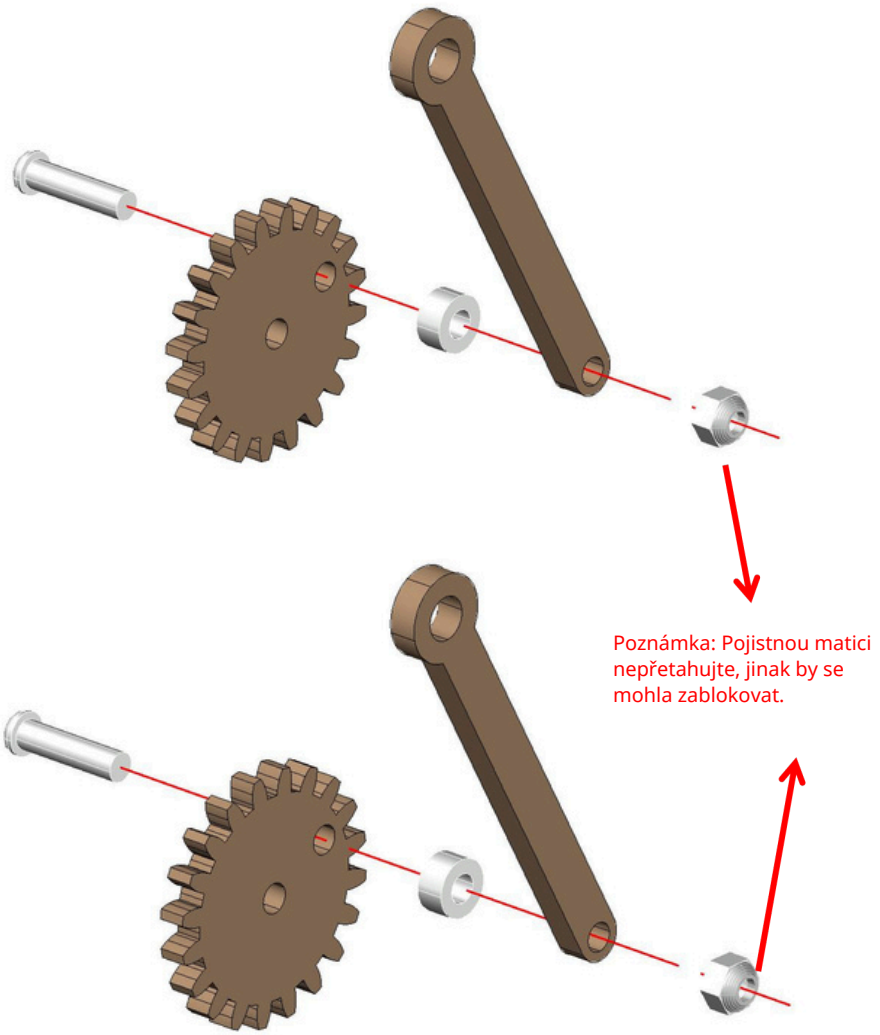
Obraz	Jméno	Množství
	Deska řadiče ESP32	1
	Ultrazvukový senzor	1

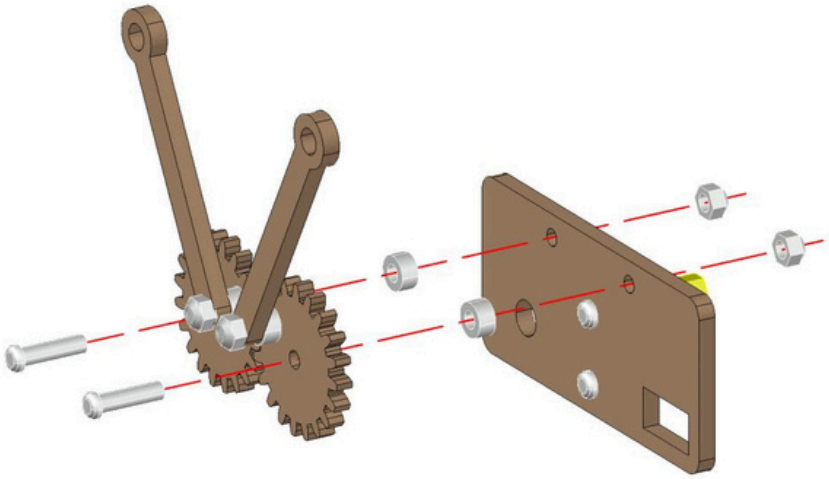
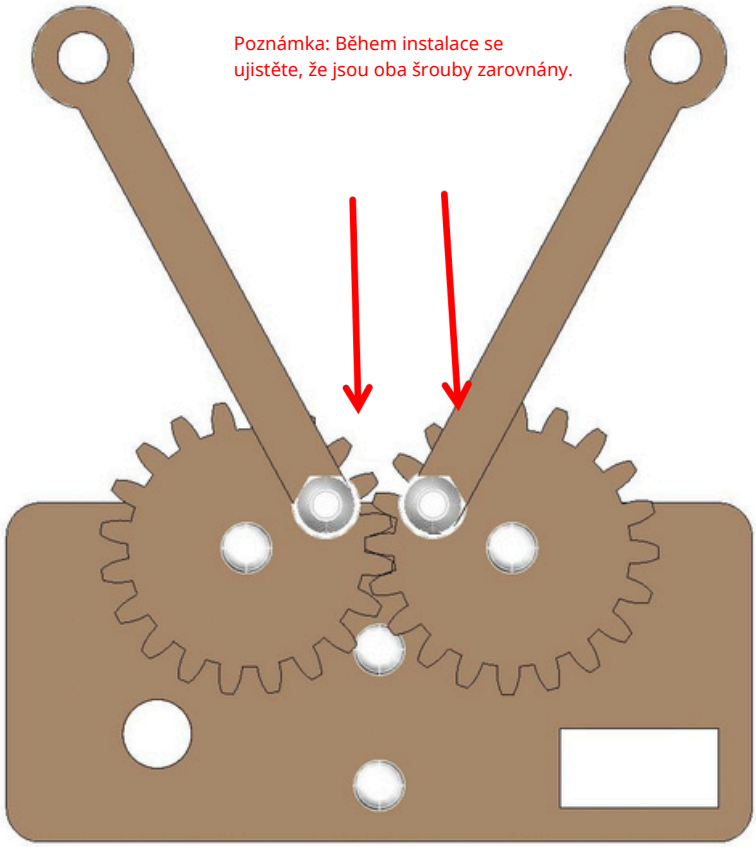
	Ovladač stejnosměrného motoru	1
	Stejnosměrný motor	1

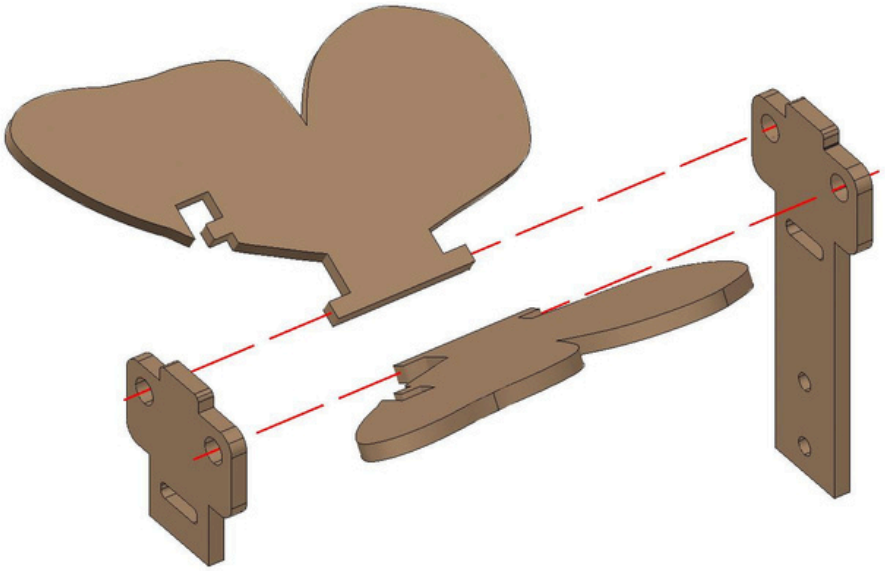
3. Sestavte bionického motýla (pokud jste si nezakoupili stavební materiál, můžete krok montáže přeskočit a přejít přímo k připojení hardwaru a učení programu)

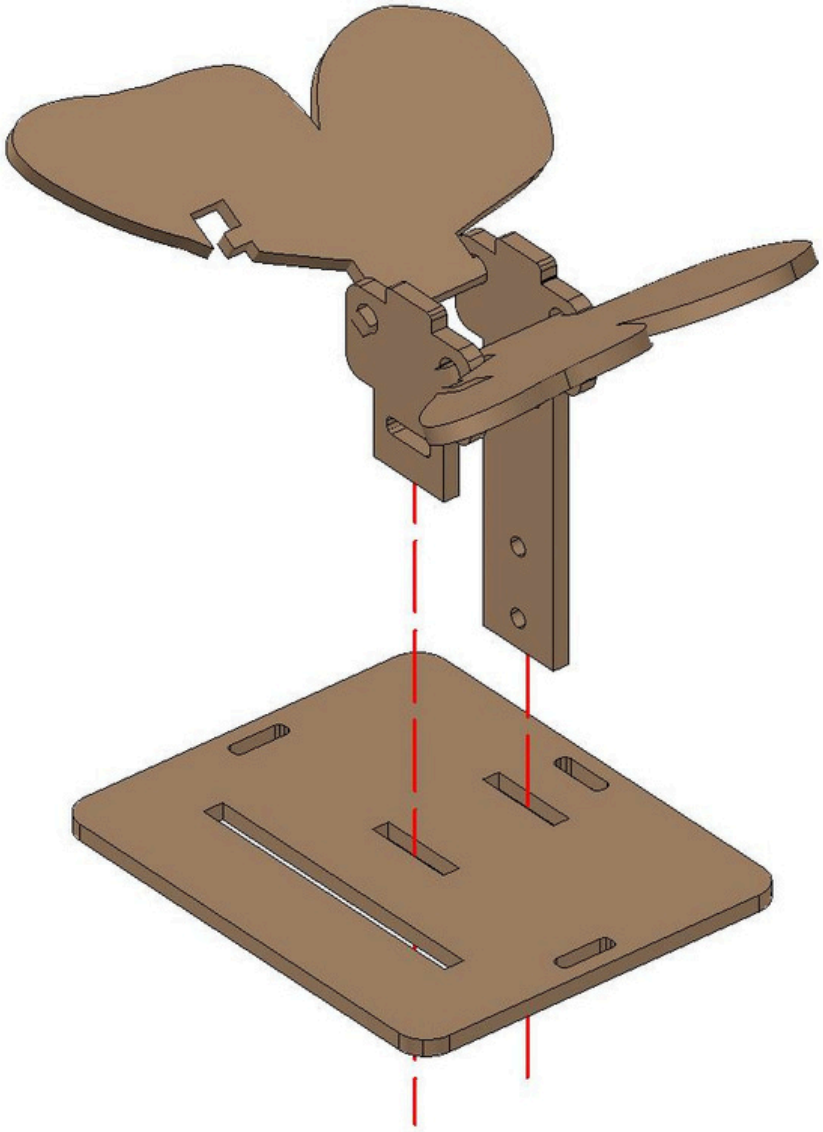
Krok 1 Instalace ultrazvukového modulu		
Seznam dílů	Ultrazvukový modul*1	Řada Dupont samec-samice*4
Ilustrace	 Připojte vodiče DuPont k pinům ultrazvukového modulu: červený vodič k VCC, modrý vodič k Trig, bílý vodič k Echo a černý vodič k GND. 	

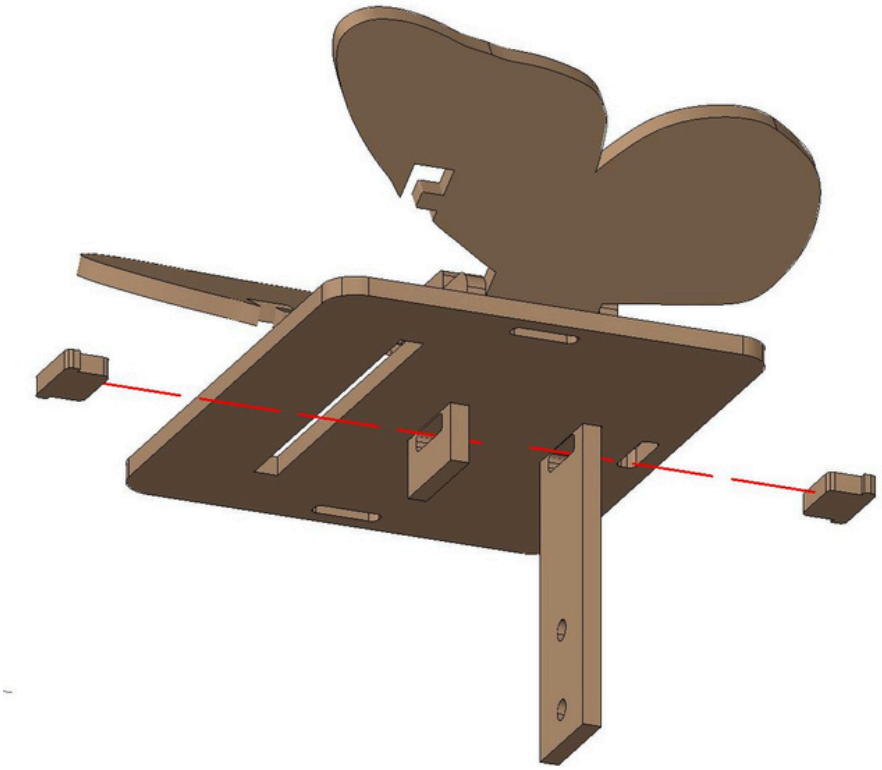
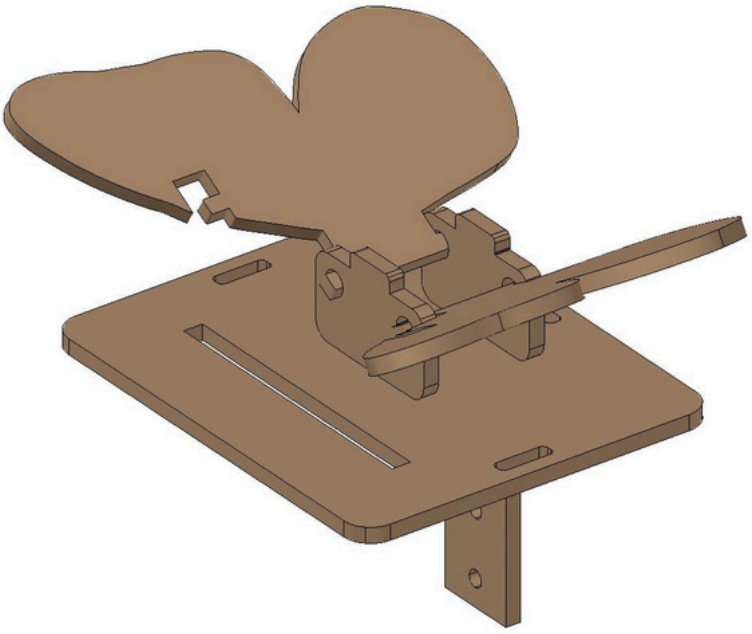
Krok 2 Instalace dvouprůchodového měděného sloupku		
Seznam dílů	M3*10MM kulatá hlava Šrouby*2	M3*12MM Dvouprůchodová měď Pilíř*2
Ilustrace		

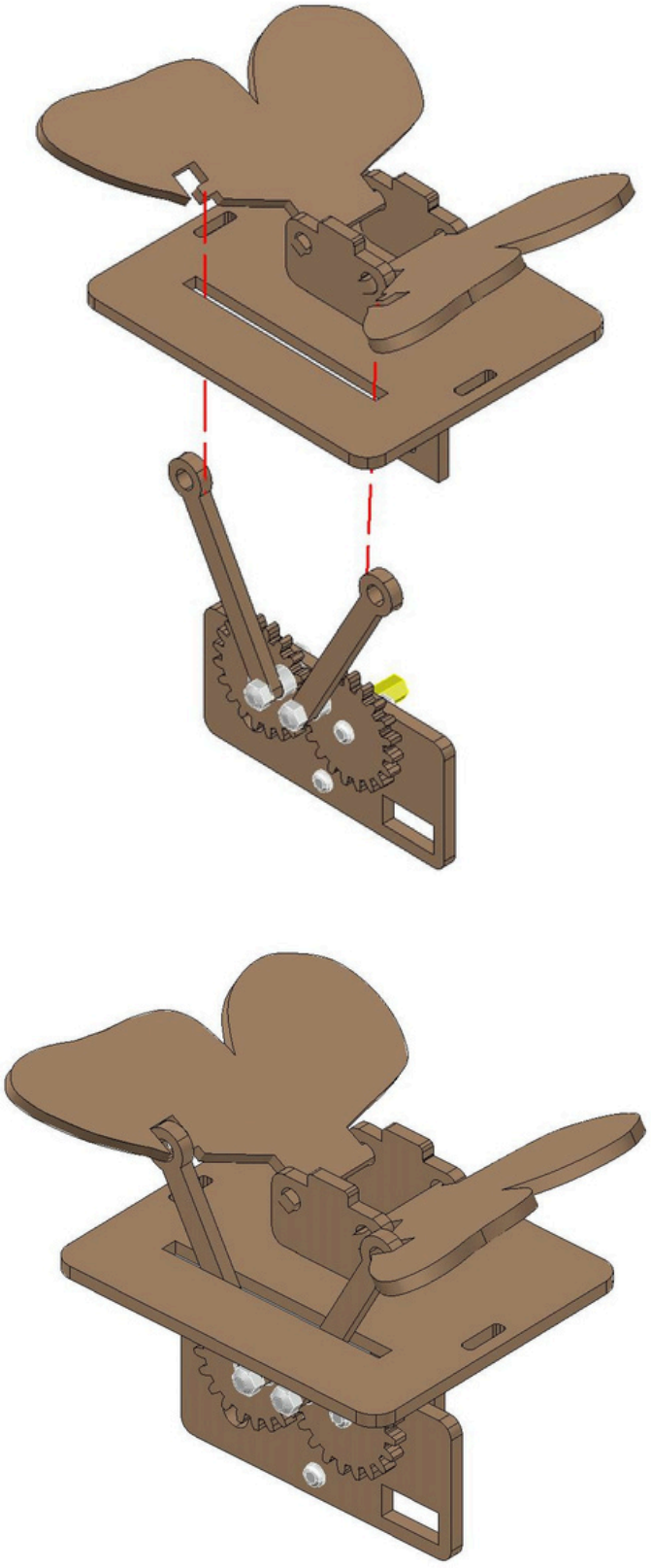
Krok 3 Sestavte konstrukci klikové hřídele a ojnice			
Seznam dílů	M3*14MM Plochá hlava Šrouby*2	M3 Poniklované Pojistné matice*2	3*7*3MM nylon Těsnění*2
Ilustrace			

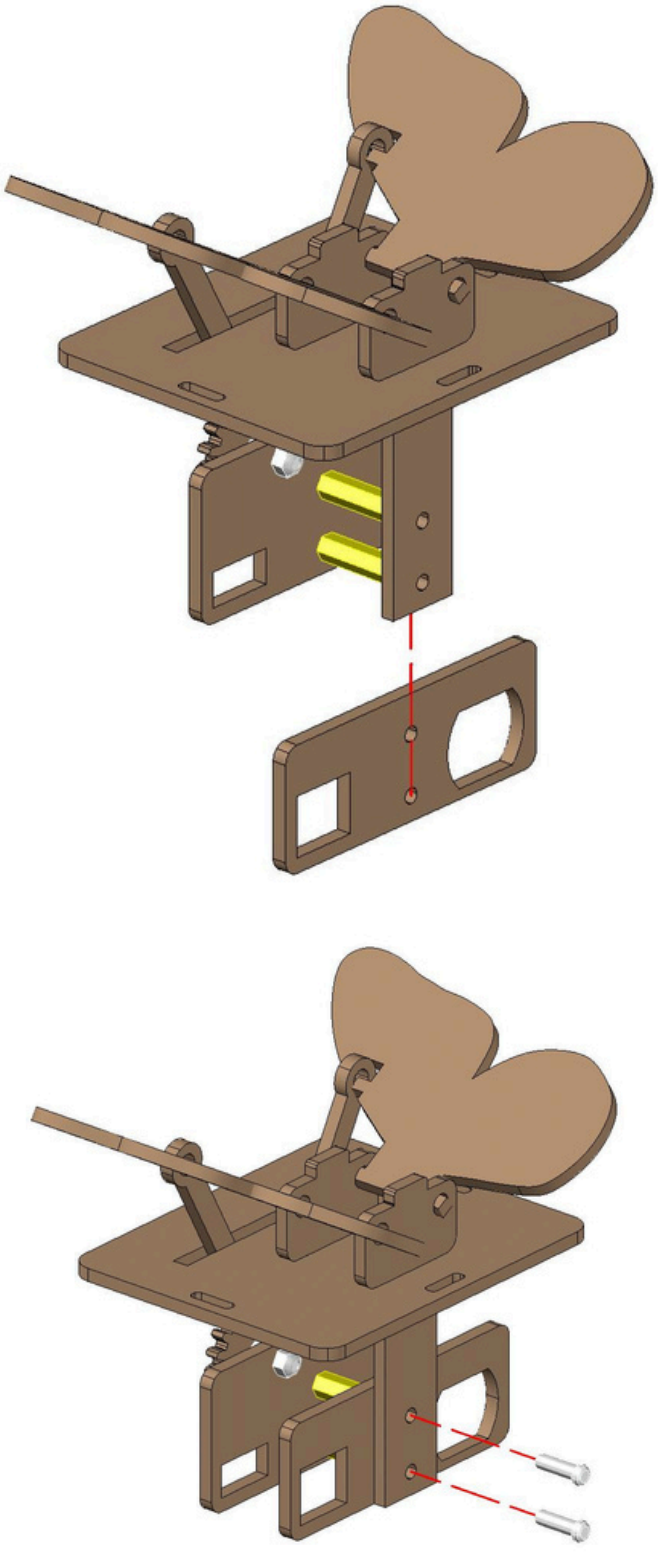
Krok 4 Nainstalujte konstrukci klikové hřídele a ojnice			
Seznam dílů	M3*14MM Plochá hlava Šrouby*2	M3 Poniklované Pojistné matice*2	3*7*3MM nylon Těsnění*2
Ilustrace	 		

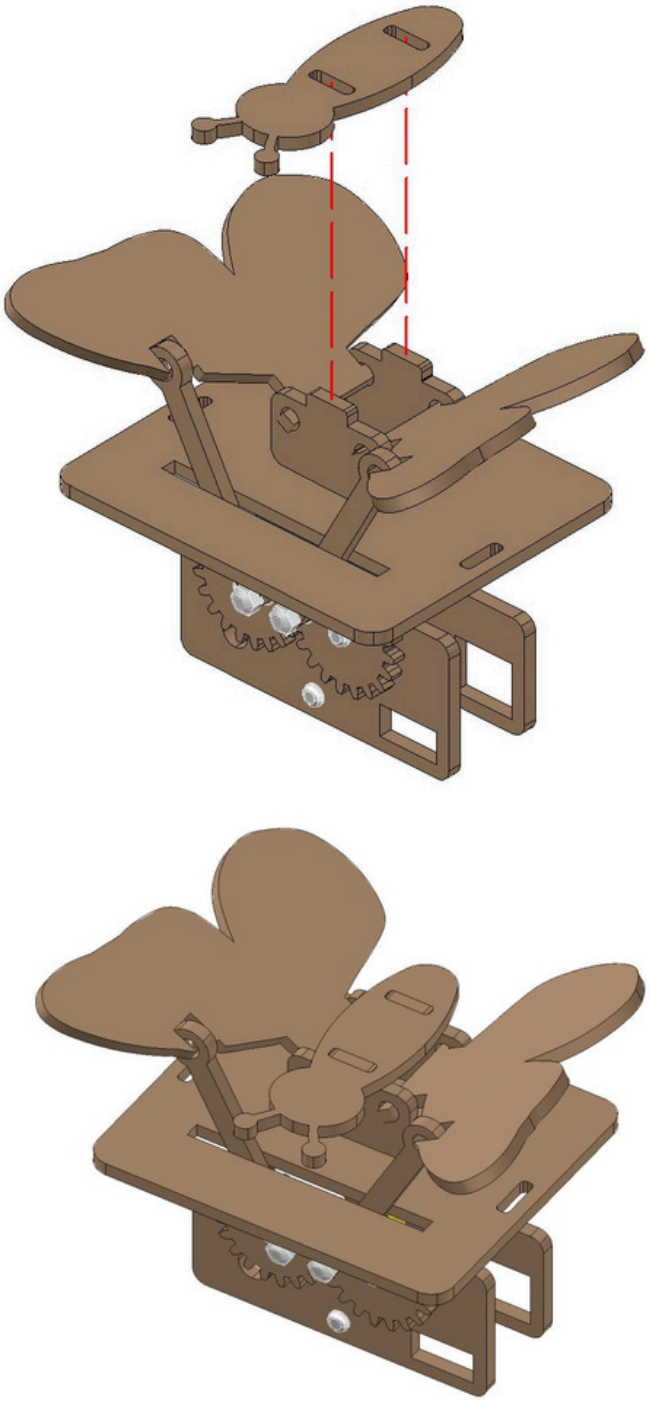
Krok 5. Instalace motýlích křídel	
Ilustrace	

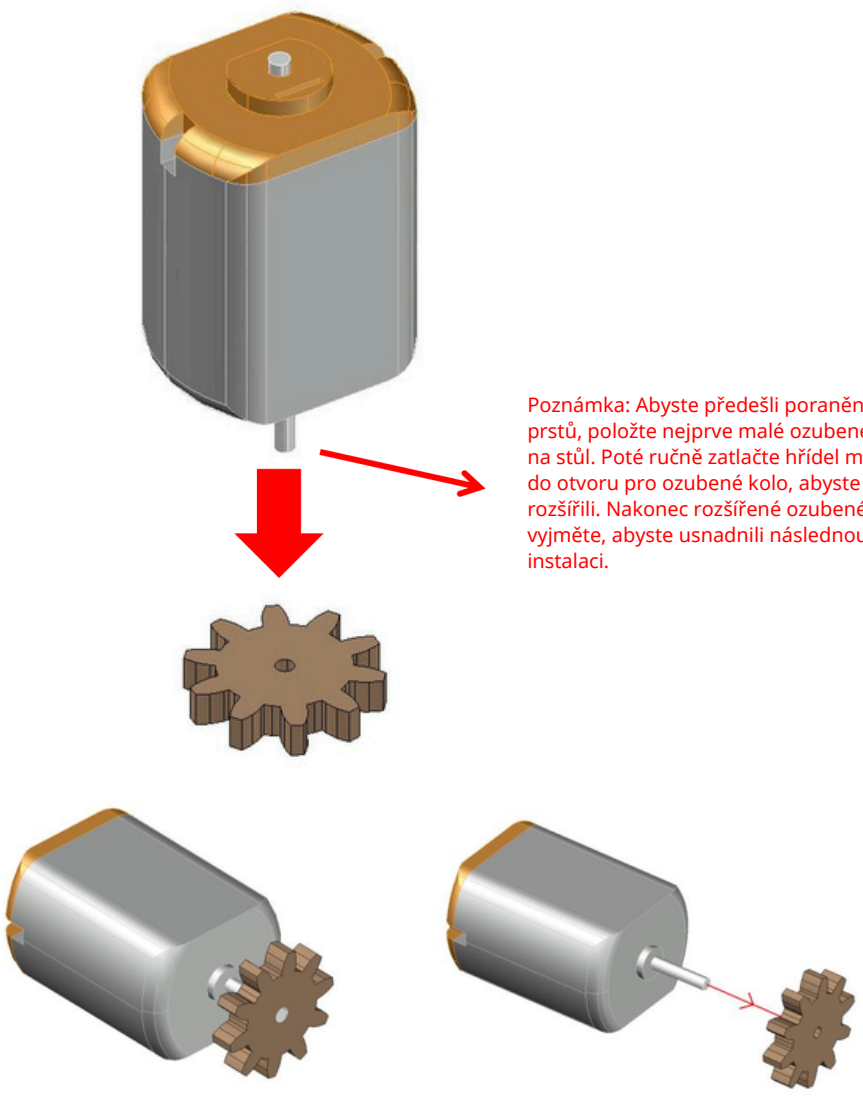
Krok 6. Zajistěte rám těla motýla	
Ilustrace	

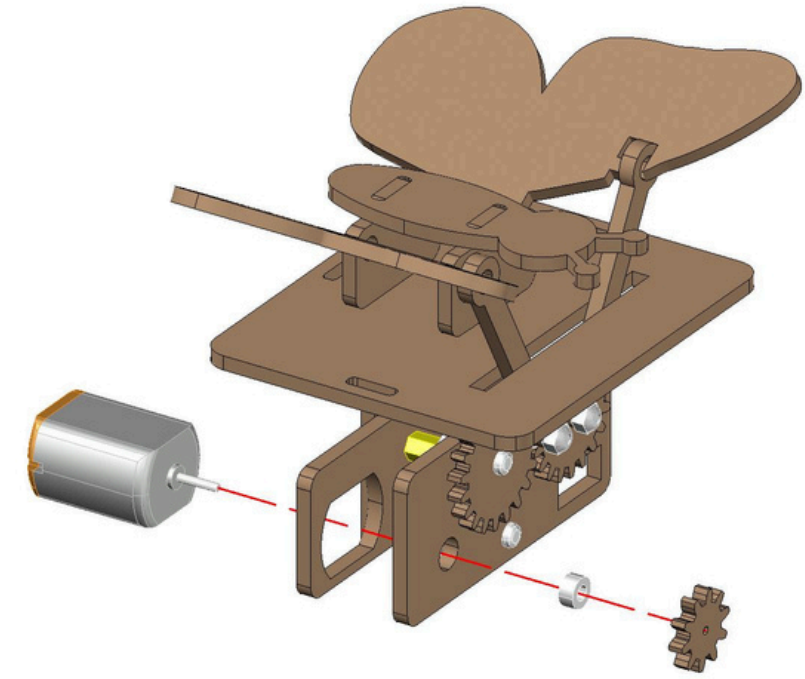
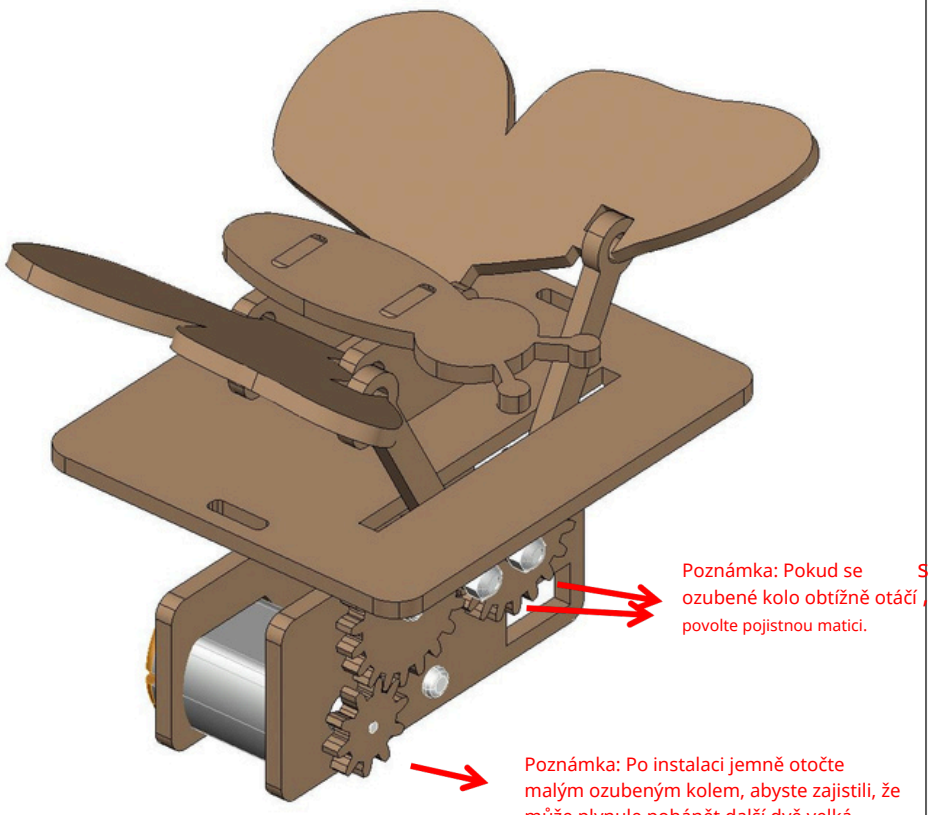
Krok 7. Zajistěte rám těla motýla	
Seznam dílů	Západka*2
Ilustrace	 

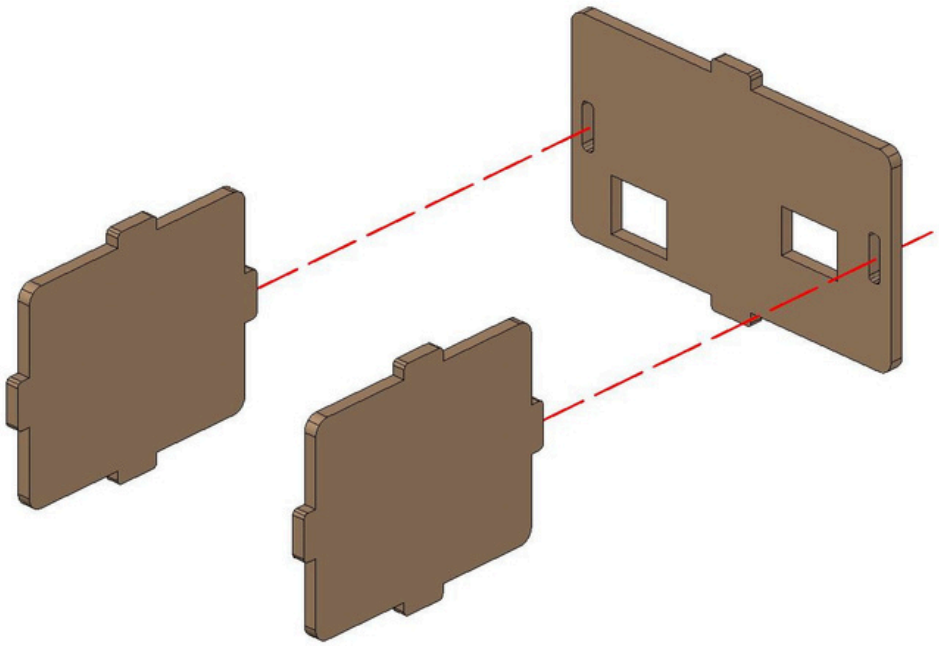
Krok 8. Zajistěte motýlí křídla	
Ilustrace	

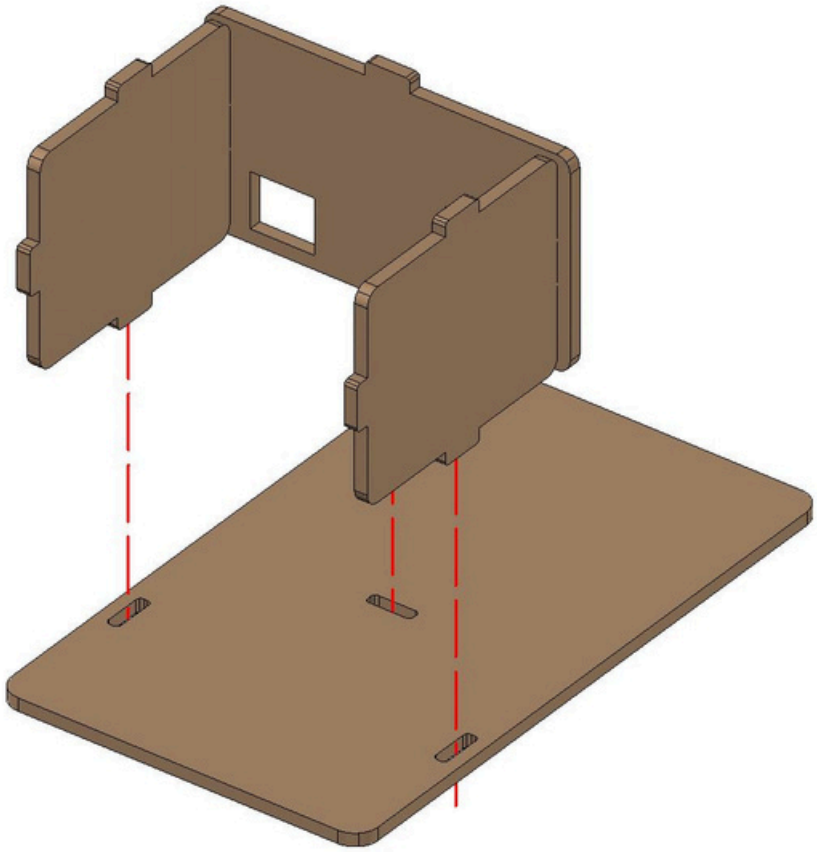
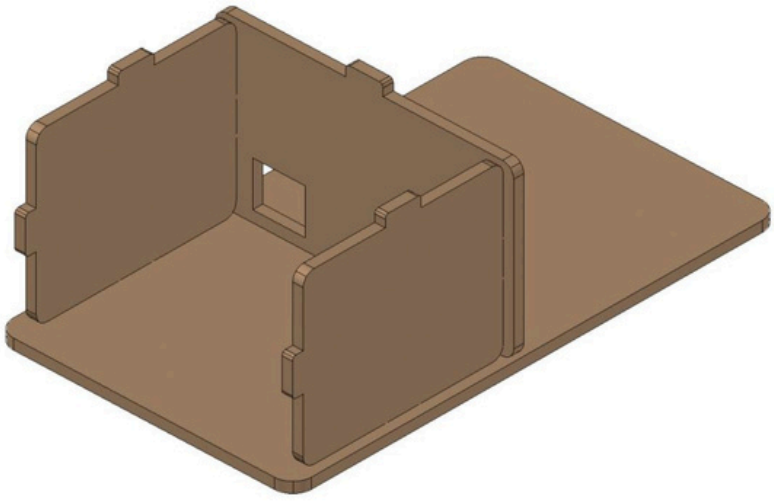
Krok 9 Nainstalujte upevňovací desku motoru	
Seznam dílů	Šrouby s kulatou hlavou M3*10MM*2
Ilustrace	

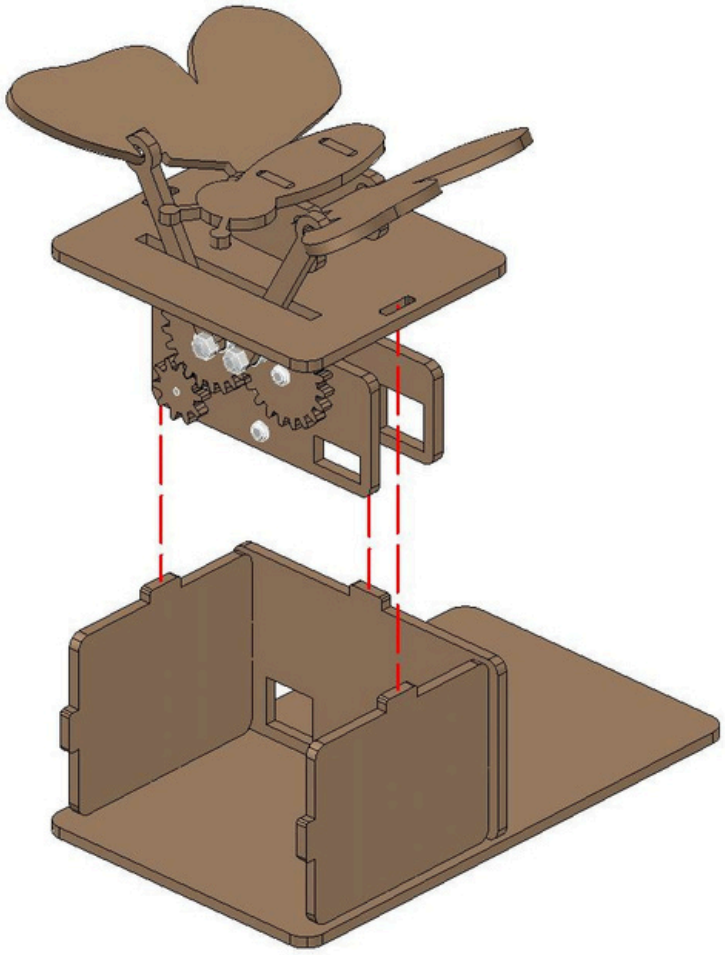
Krok 10. Instalace těla motýla	
Ilustrace	

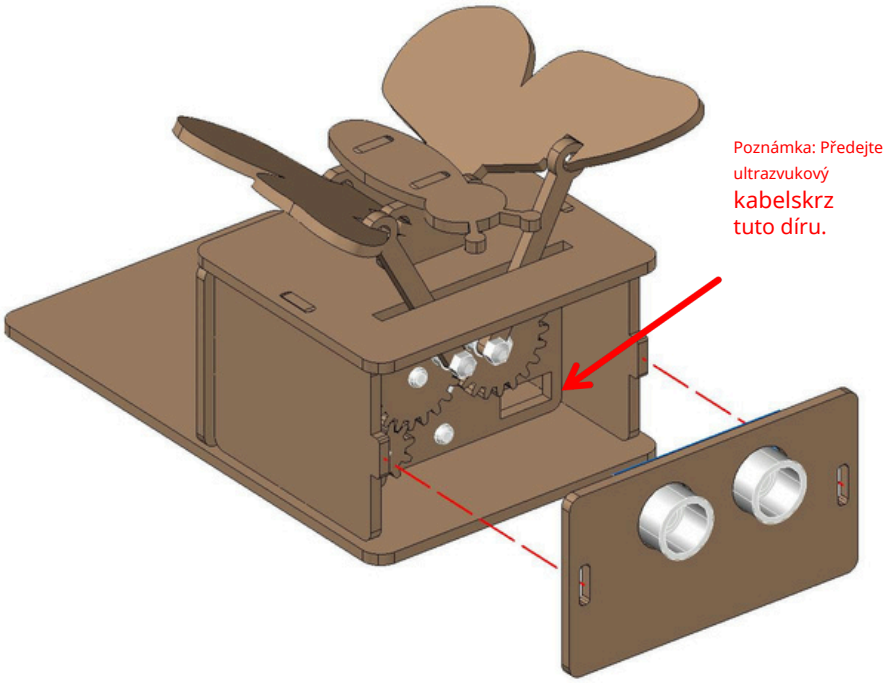
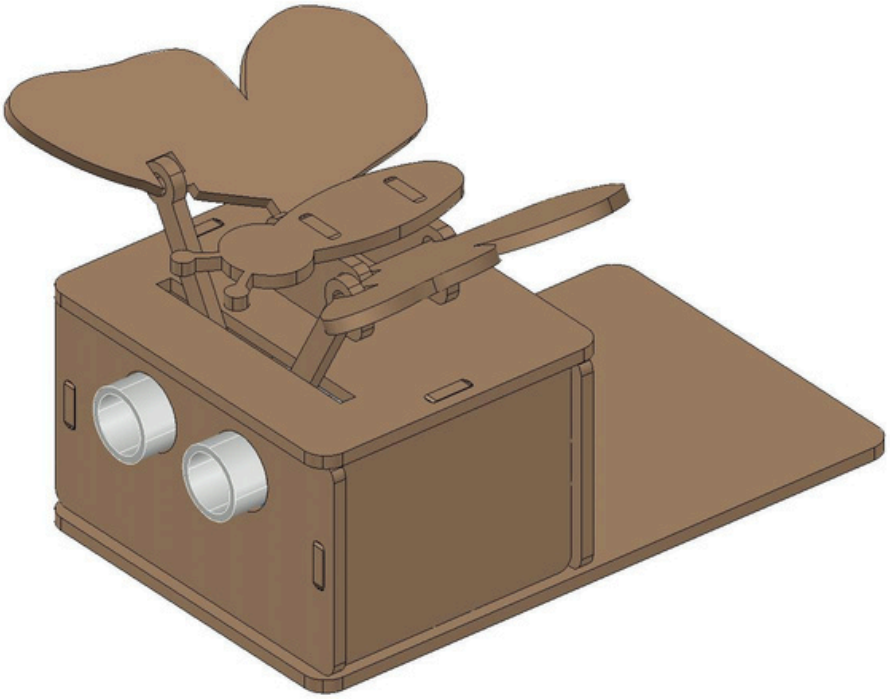
Krok 11. Rozšíření otvoru pastorku	
Ilustrace	

Krok 12 Instalace motoru	
Seznam dílů	Motor*1
Ilustrace	 

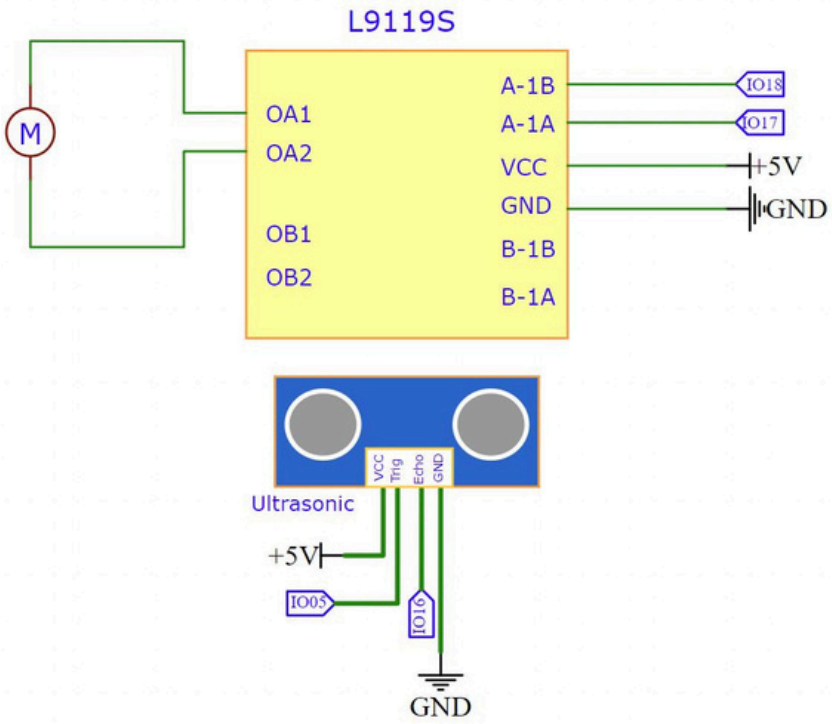
Krok 13 Instalace bočních panelů	
Ilustrace	

Krok 14 Instalace spodního panelu	
Ilustrace	 

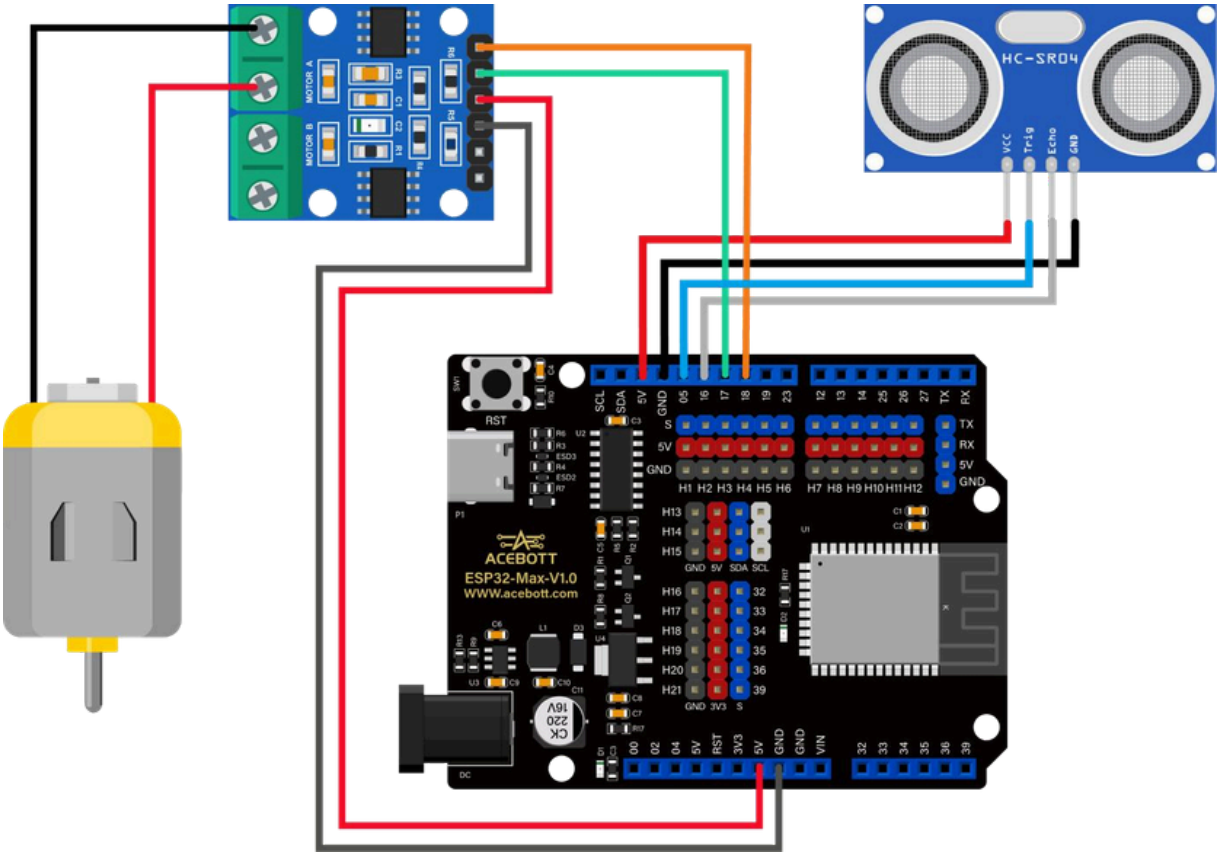
Krok 15. Zajištění těla motýla	
Ilustrace	

Krok 16 Instalace předního panelu	
Ilustrace	 

4. Schéma zapojení



5. Připojení hardwaru



6. Referenční postupy Otevřete "Smart_Bionic_Butterfly.ino „\Arduino(Zkušený student)\2.Code\lesson 34 Smasorutbo rBionic Butterfly\Smart_Bionic_Butterfly“ nebo zkopírujte následující kód do Arduino IDE. Po výběru vývojové desky (ESP32 Dev Module) a příslušného portu podle dříve naučených kroků klikněte na

nahrát program.

```
const int trigPin = 5; // Trigger pin of ultrasonic sensor
const int echoPin = 16; // Echo pin of ultrasonic sensor
const int motor1A = 17; // Motor direction control pin
const int motor1B = 18; // PWM speed control pin
const float thresholdDistance = 20.0;  // Distance threshold in centimeters

void setup() {
Serial.begin(115200); // Initialize serial communication
pinMode(trigPin, OUTPUT); // Set trigger pin as output
pinMode(echoPin, INPUT); // Set echo pin as input
pinMode(motor1A, OUTPUT); // Set motor direction pin as output
pinMode(motor1B, OUTPUT); // Set motor speed pin as output

}

void loop() {
float distance = getDistance(); // Measure distance
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");
// If object is detected within the threshold distance, run motor
if (distance != -1 && distance < thresholdDistance) {
analogWrite(motor1B, 255); // Set direction
analogWrite(motor1A, 0); // Set PWM speed
} else if (distance != -1 && distance >= thresholdDistance) {
analogWrite(motor1B, 0); // Set direction
analogWrite(motor1A, 0); // Set PWM speed
}
delay(100);  // Short delay for better response
}

// Measure distance using ultrasonic sensor
float getDistance() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

unsigned long pulseTime = pulseIn(echoPin, HIGH, 5000); // Timeout after
5000 µs

if (pulseTime > 0) {
return pulseTime * 0.0343 / 2; // Convert pulse duration to distance (cm)
} else {
return -1;  // Return -1 if timeout (no object detected)
}
}
```

Po úspěšném nahrání programu začněte mávat křídly, když je někdo detekován do vzdálenosti 20 cm přímo před vámi.

Lekce 35 Chytrý odpadkový koš

Cíle kurzu

- Naučte se, jak fungují chytré odpadkové koše a jaké jsou scénáře použití.
- Naučte se sestavit model chytrého odpadkového koše.

-Napište program, který implementuje funkci automatického otevírání víka inteligentních odpadkových košů.

Úvod do kurzu

V každodenním životě se hygiena odpadkových košů často opomíjí. Chytrý odpadkový koš pomocí senzorů snímá blízkost lidských rukou nebo předmětů, automaticky otevírá víko, zabraňuje přímému kontaktu a účinně zlepšuje hygienu a snadnost používání. S rozvojem chytrých domácností se navrhuje stále více chytrých zařízení a chytré odpadkové koše jsou jedním z nich.

V této lekci se naučíme, jak si postavit praktický chytrý odpadkový koš.

Body znalostí

1. Jak funguje chytrý odpadkový koš?

Chytré odpadkové koše obvykle používají infračervené nebo ultrazvukové senzory k detekci přiblížení osoby nebo předmětu. Pokud je detekovaná vzdálenost menší než nastavená prahová hodnota, servopohon ovládá víko tak, aby se otevřelo a automaticky zavřelo, když se předmět vzdálí. Tím se dosáhne „bezkontaktního“ provozu, což zvyšuje pohodlí a hygienu.

Experiment

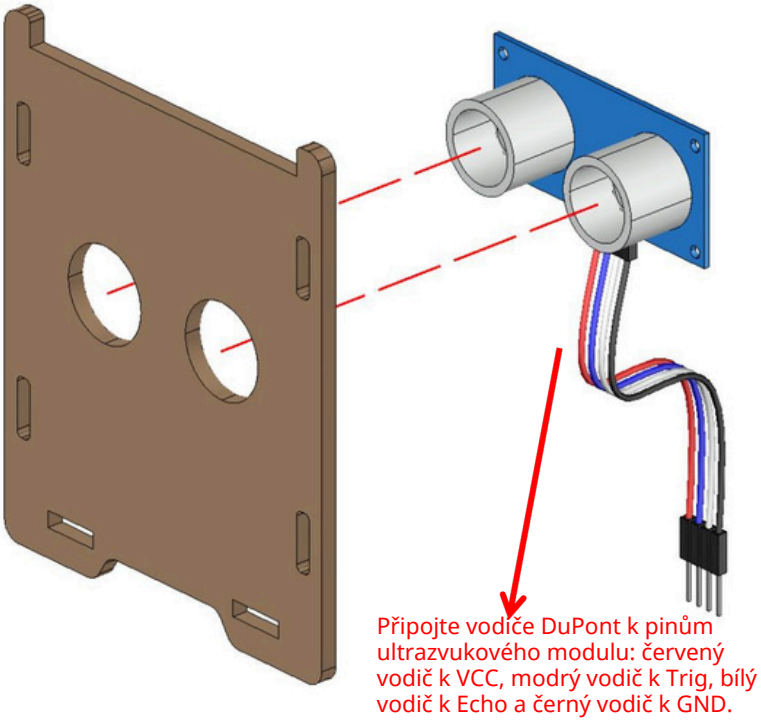
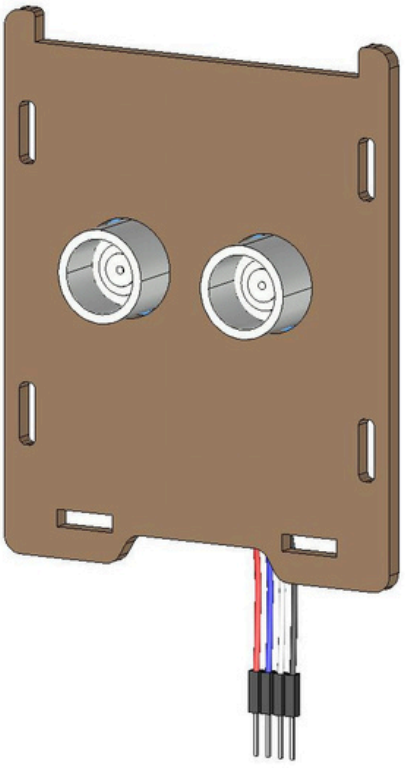
1. Popis projektu

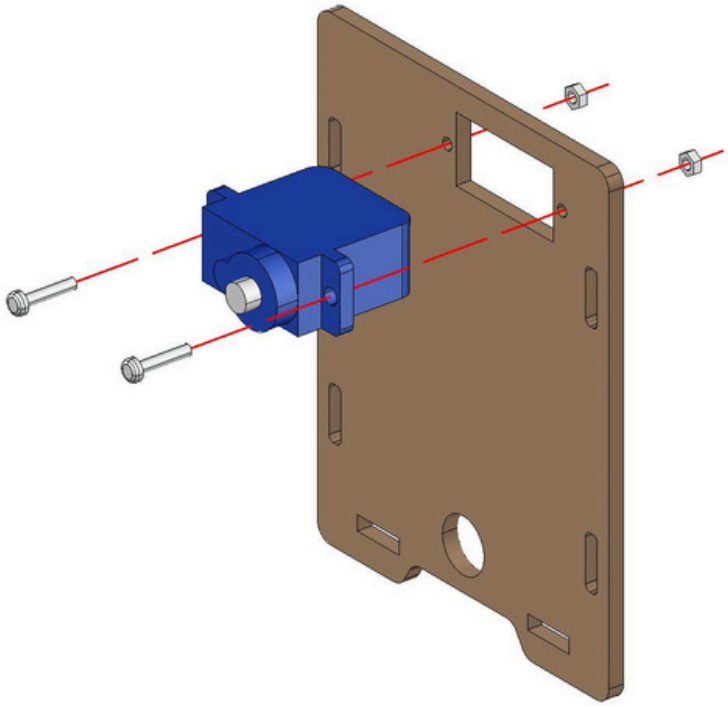
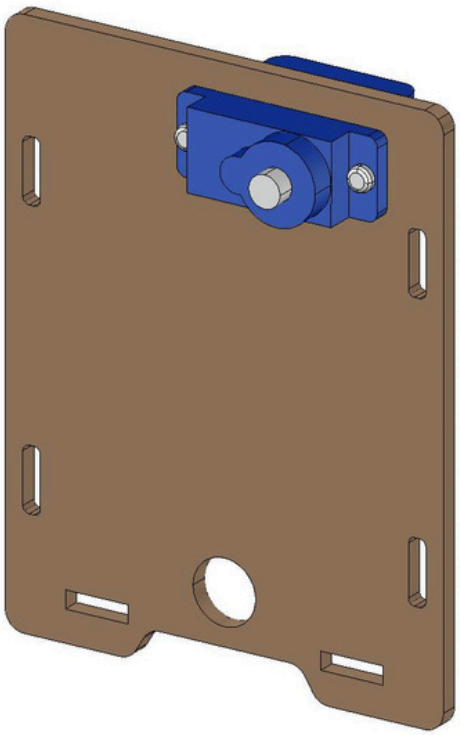
Tento program využívá ultrazvukový senzor k detekci blízkosti člověka. Když je někdo detekován v dosahu 25 centimetrů, servomotor automaticky otevře víko koše. Když se objekt vzdálí, servomotor zpozdí zavření víka, čímž vzniká bezkontaktní inteligentní systém koše.

2. Seznam hardwaru

Obraz	Jméno	Množství
	Deska řadiče ESP32	1
	Ultrazvukový senzor	1
	Servo	1

3. Sestavte chytrý odpadkový koš (pokud jste si nezakoupili stavební materiál, můžete krok montáže přeskočit a přejít přímo k připojení hardwaru a učení programu)

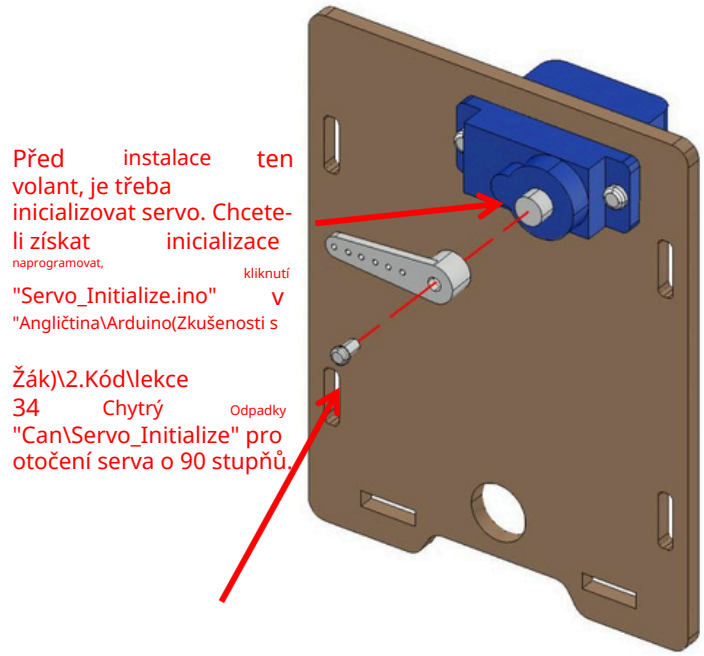
Krok 1 Instalace ultrazvukového modulu		
Seznam dílů	Ultrazvukový modul*1	Řada Dupont samec-samice*4
Ilustrace	 	

Krok 2 Instalace serva			
Seznam dílů	Servo *1	M2*14MM kulatá hlava Šroub*2	Matice M2*2
Ilustrace	 		

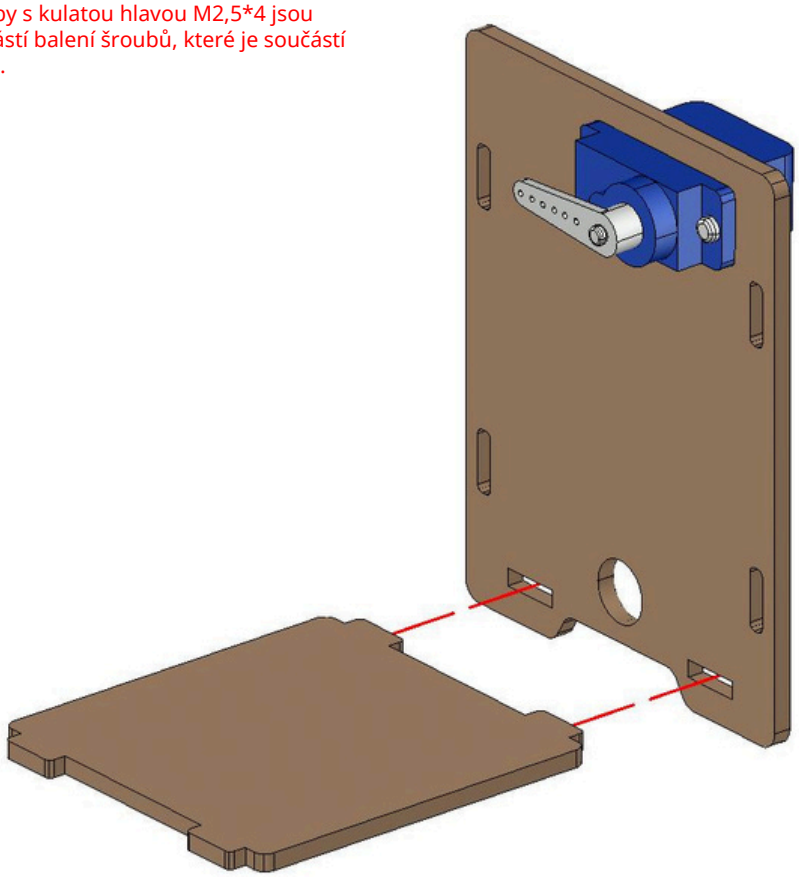
Krok 3. Nainstalujte volant a spodní desku

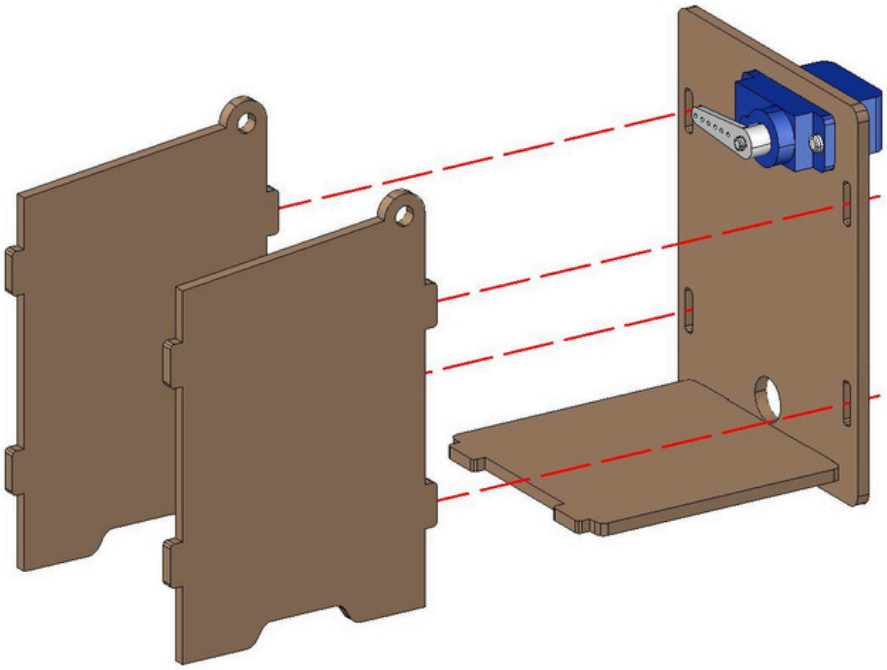
Seznam dílů	Servo klakson*1	Šrouby s kulatou hlavou M2,5*4*1

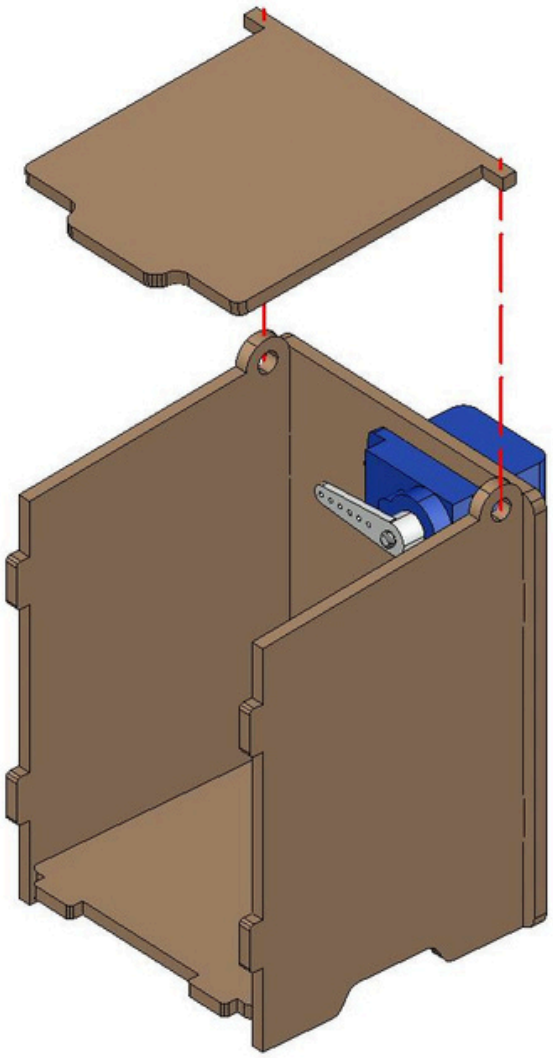
ilustrace

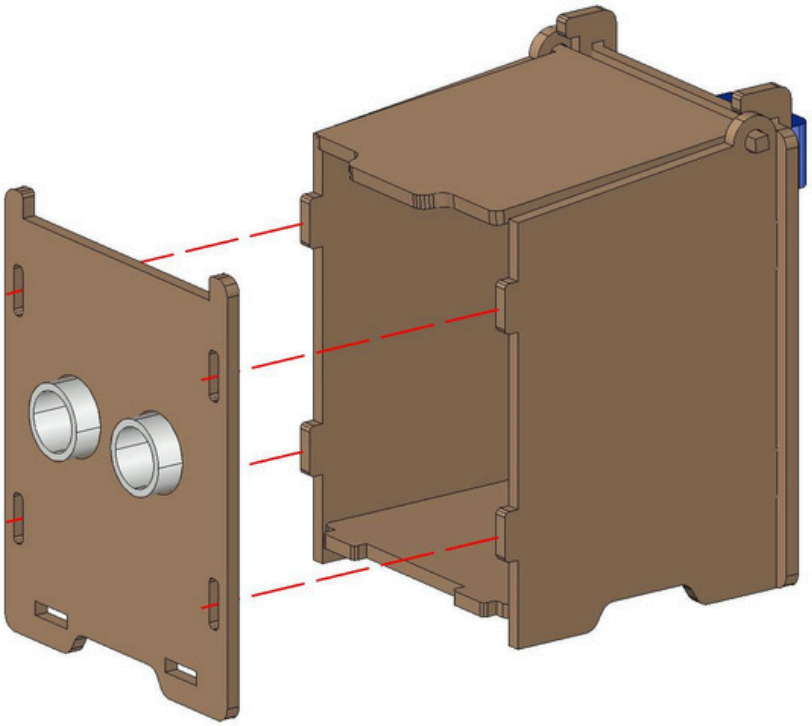


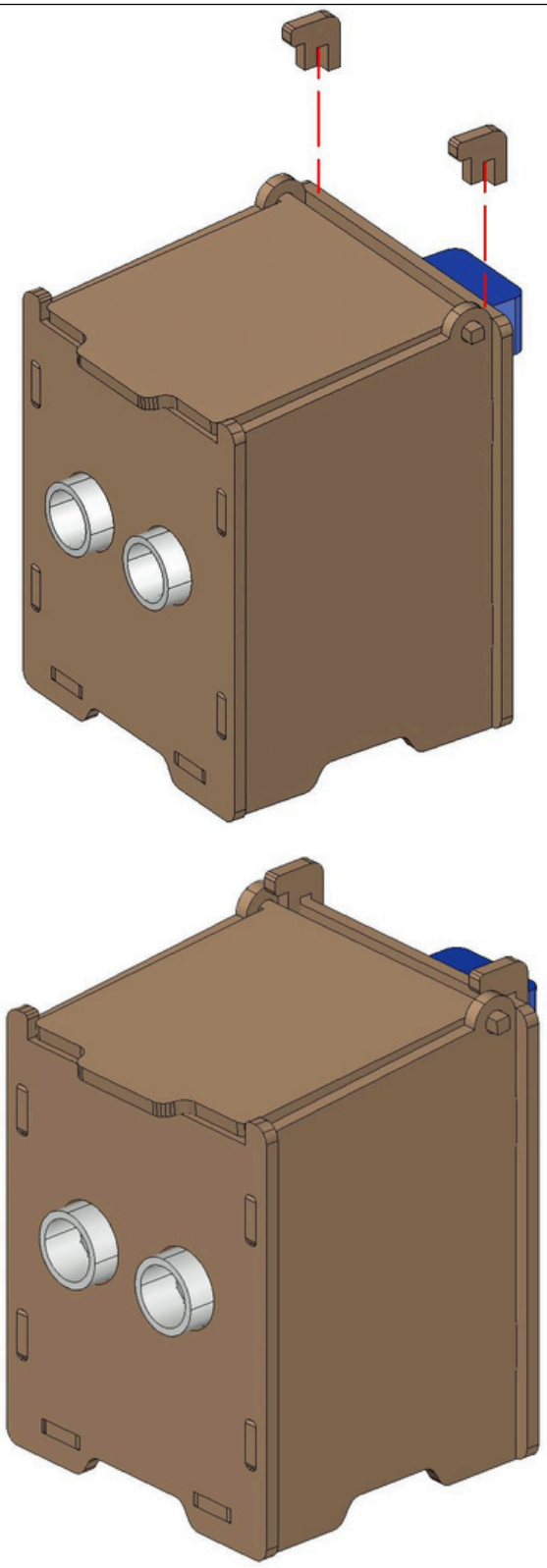
Šrouby s kulatou hlavou M2,5*4 jsou součástí balení šroubů, které je součástí serva.



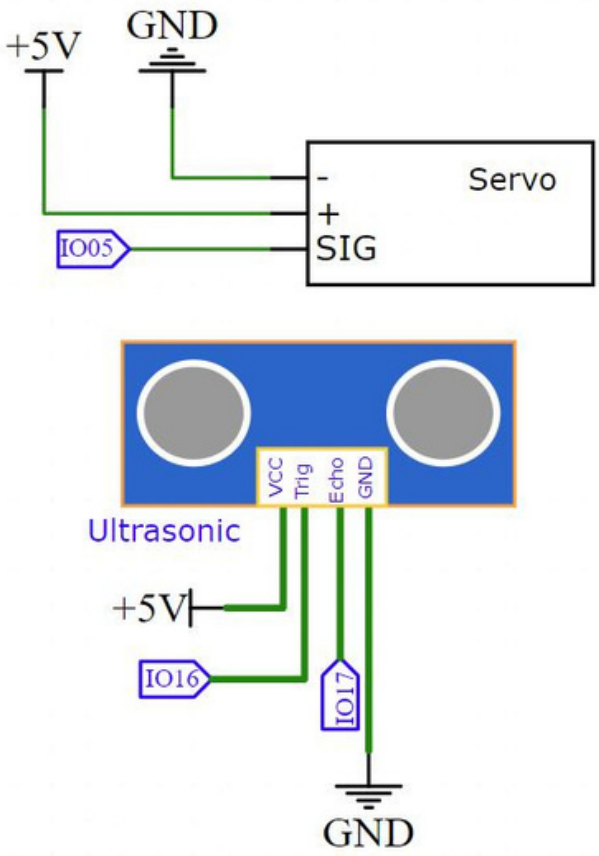
Krok 4. Instalace bočních panelů odpadkového koše	
Ilustrace	

Krok 5. Instalace krytu odpadkového koše	
Ilustrace	

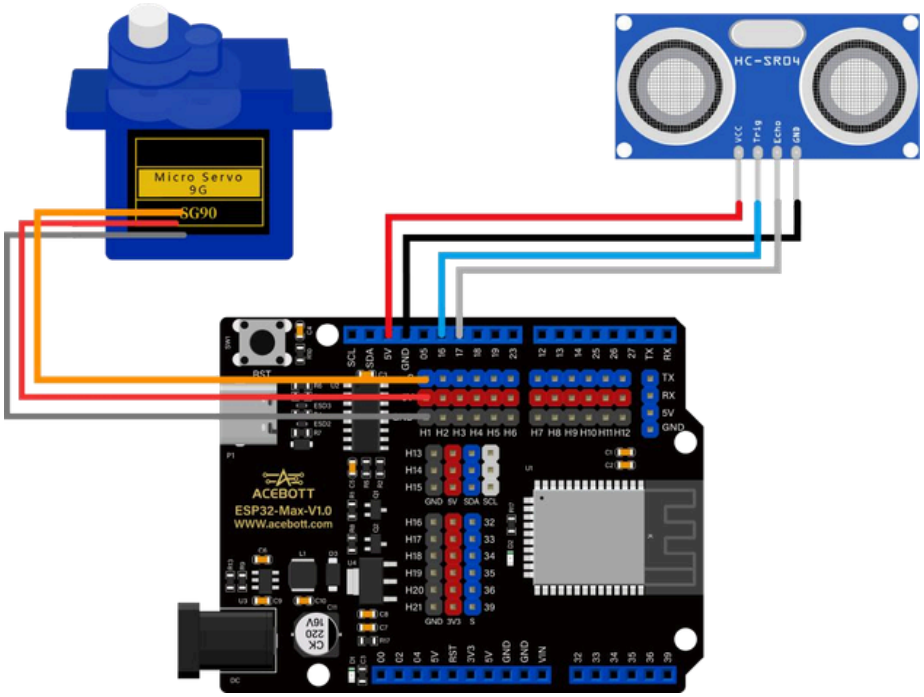
Krok 6. Instalace předního panelu odpadkového koše	
Ilustrace	

Krok 7. Instalace omezovače krytu odpadkového koše	
Ilustrace	

4. Schéma zapojení



5. Připojení hardwaru



6. Referenční postupy

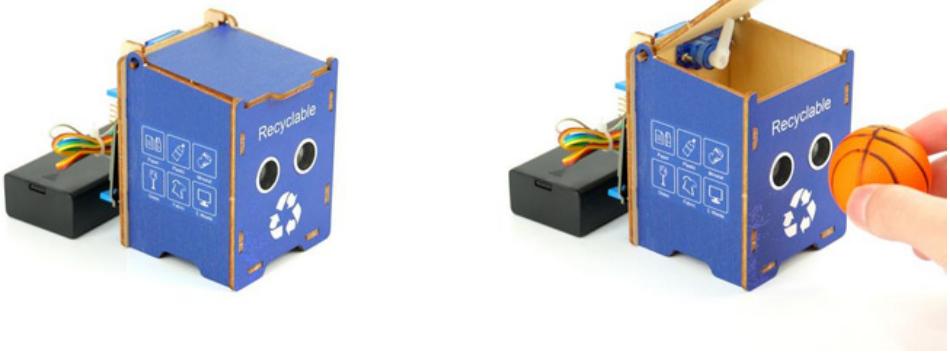
Otevřete "Smart_Trash_Can.ino "soubor v "English\Arduino(Experienced

Learner)\2.Code\lesson 35 Chytrý odpadkový koš\Smart_Trash_Can" nebo zkopírujte následující kód do Arduino IDE. Po výběru vývojové desky (ESP32 Dev Module) a příslušného portu podle pokynů dříve naučené kroky, klikněte  nahrát program.

```
#include <ESP32Servo.h>

#define TRIG_PIN 16         // Ultrasonic sensor trigger pin
#define ECHO_PIN 17         // Ultrasonic sensor echo pin
#define SERVO_PIN 5         // Servo control pin

int angle = 90;             // Initial servo angle (closed position)
bool isOpened = false;      // Flag to indicate if the lid is currently open

Servo myServo;

// Function to read distance from ultrasonic sensor (in centimeters)
float readDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH); // Timeout after 30ms
  long distance = duration * 0.034 / 2;           // Convert time to distance
  delay(100);                                       // Small delay to
stabilize reading
  return distance;
}

void setup() {
  Serial.begin(115200);           // Initialize serial monitor
  pinMode(TRIG_PIN, OUTPUT);      // Set trigger pin as output
  pinMode(ECHO_PIN, INPUT);       // Set echo pin as input
  myServo.setPeriodHertz(50);     // Standard servo frequency: 50Hz
  myServo.attach(SERVO PIN, 500, 2500); // Set servo pulse width range
(500-2500µs)
  myServo.write(angle);           // Move servo to initial (closed)
position
}

void loop() {
  float distance = readDistance();  // Get distance measurement
  Serial.print("Distance: ");
  Serial.print(distance);
```

```
  Serial.println(" cm");

  // If an object is detected within 25 cm and lid is closed, open it
  if (distance > 0 && distance < 25 && isOpened == false) {
    for (angle = 90; angle < 170; angle++) {
      myServo.write(angle);       // Gradually open the lid
      delay(5);                   // Small delay for smooth motion
    }
    isOpened = true;              // Update state flag
  }
  // If object is far and lid is open, close it
  else if (distance >= 25 && isOpened == true) {
    delay(500);                   // Wait 0.5 seconds before closing
    for (angle = 170; angle > 90; angle--) {
      myServo.write(angle);       // Gradually close the lid
      delay(5);                   // Small delay for smooth motion
    }
    isOpened = false;             // Update state flag
  }
  delay(200);                     // Short delay before next reading
}
```

Po úspěšném nahrání programu se chytrý odpadkový koš otevře, když detekuje přicházející osobu, a zavře se, když nikdo detekován není.



Lekce 36 Chytrá brána

Cíle kurzu

- Zjistěte, jak fungují chytré turnikety
- Sestavte a postavte inteligentní model turniketu
- Napište program pro implementaci funkce inteligentního turniketu

Úvod do kurzu

Už jste někdy viděli u vjezdu do nákupního centra, kancelářské budovy nebo veřejného parkoviště chytrou bránu, která automaticky zvedá zábradlí, když se blíží auto? Toto zařízení nevyžaduje ruční ovládání a dokáže automaticky detekovat a otevřít zábradlí podle blízkosti vozidla, což výrazně zlepšuje efektivitu dopravy a úroveň inteligence. Jak tedy „ví“, že se blíží auto? Jak ovládat zvedání nebo spouštění zábradlí?

V této lekci si postavíme model inteligentní automobilové brány. Použijeme ultrazvukové senzory pro snímání vozidel a servopohon pro řízení pohybu.

bariéry, zažijeme praktické využití inteligentních systémů v reálném provozu.

Body znalostí

1. Inteligentní brány

Inteligentní brána je inteligentní zařízení používané pro správu přístupu vozidel a automatické ovládání. Běžně se používá v místech, jako jsou obytné komplexy, parkoviště, kancelářské brány a vjezdy na dálnice. Integruje snímání, řídicí logiku a akční členy pro identifikaci, určení a uvolnění nebo zachycení vjíždějících a vyjíždějících vozidel.

Jak fungují chytré brány?

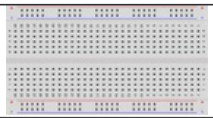
Základním principem inteligentní automobilové brány je automatizovaný proces „snímání + řízení + provedení“. Nejprve ultrazvukový senzor instalovaný v přední části brány nepřetržitě vysílá zvukové vlny a měří dobu návratu, aby určil, zda se blíží vozidlo. Jakmile je naměřená vzdálenost menší než nastavená hodnota (například 20 cm), hlavní řídicí čip (například ESP32) okamžitě reaguje a odesílá řídicí signál do serva. Po přijetí signálu se servo začne otáčet, čímž se závora zvedne a dokončí se akce „otevření brány“. Když vozidlo projede a opustí snímanou oblast, systém znovu potvrdí, že v okolí nejsou žádné překážky, a poté dá servu pokyn k pomalému spuštění závory a jejímu návratu do původního stavu. Celý proces nevyžaduje žádnou ruční obsluhu a spoléhá se na data ze senzorů a programové logické řízení pro dosažení efektivního a bezpečného automatického řízení dopravy.

Experiment

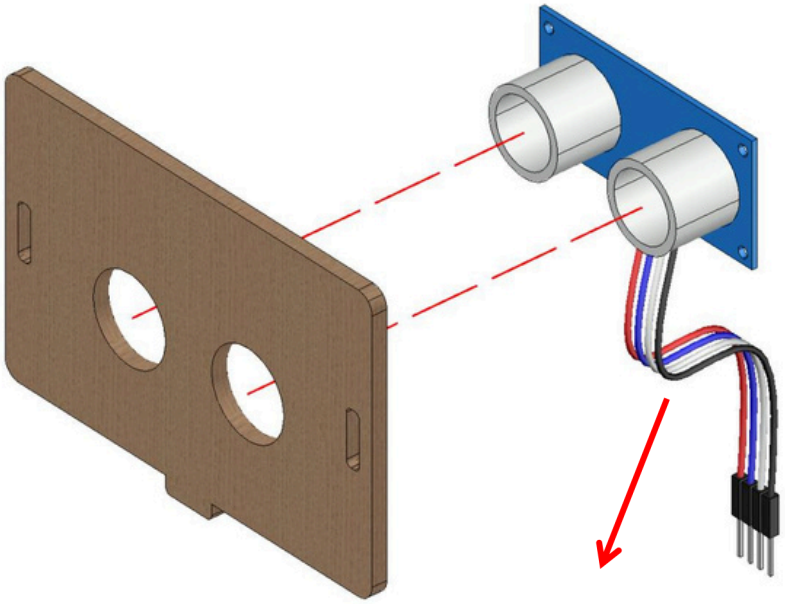
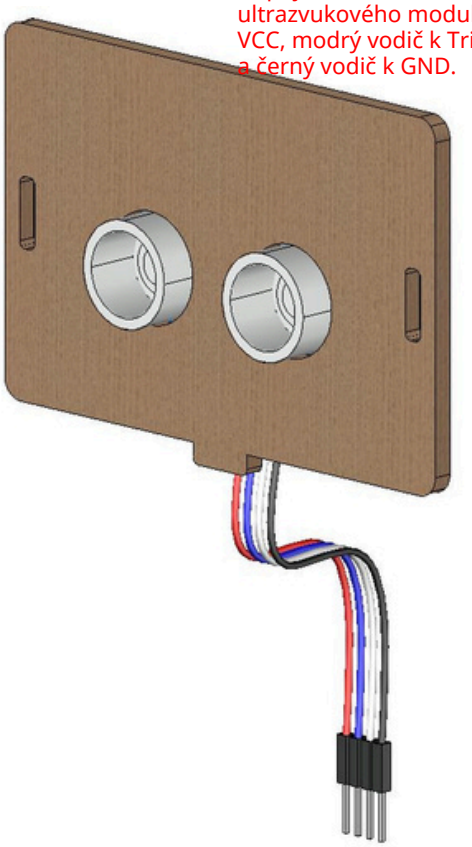
1. Popis projektu

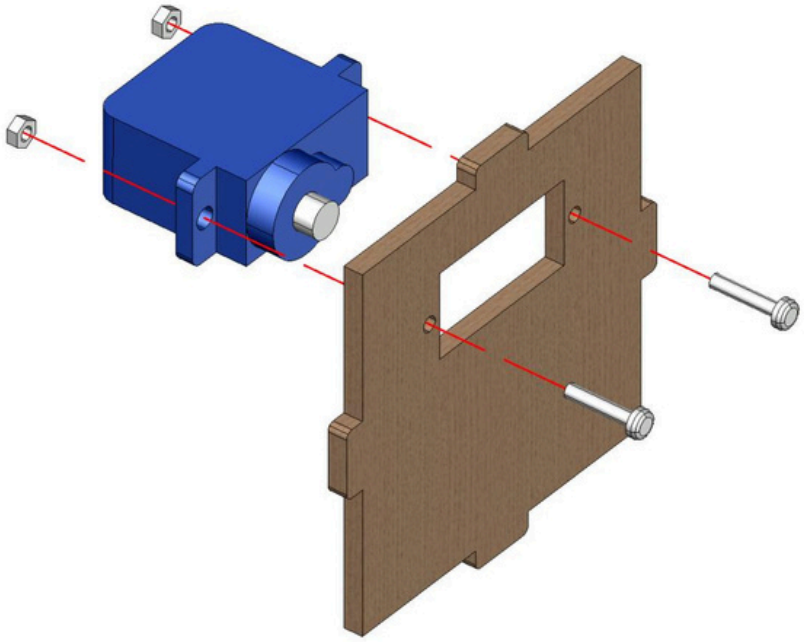
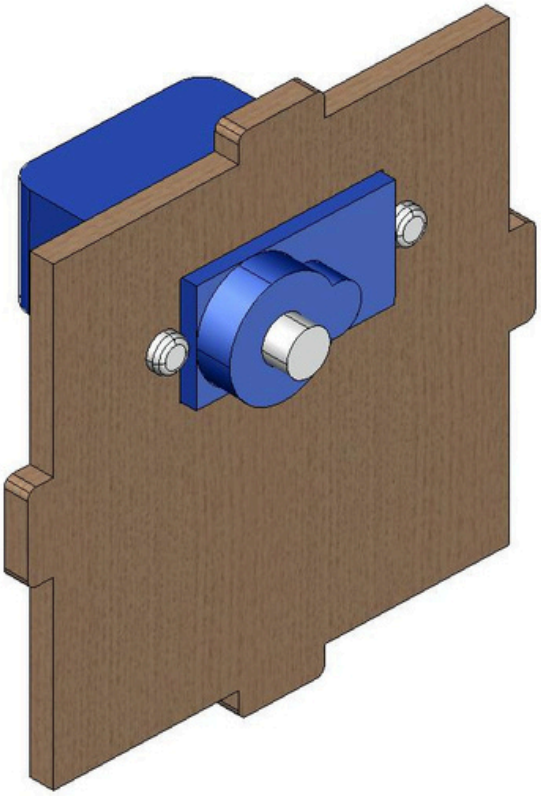
Tento program implementuje inteligentní systém brány. Když se vozidlo nebo objekt přiblíží k bráně (s detekční vzdáleností menší než 14 cm), systém pomocí serva automaticky zvedne závoru brány a rozsvítí kontrolku, která signalizuje průjezd. Když vozidlo odjede (s detekční vzdáleností větší než 20 cm), servo pomalu spustí závoru brány a zhasne kontrolku, čímž dokončí inteligentní operaci otevírání a zavírání.

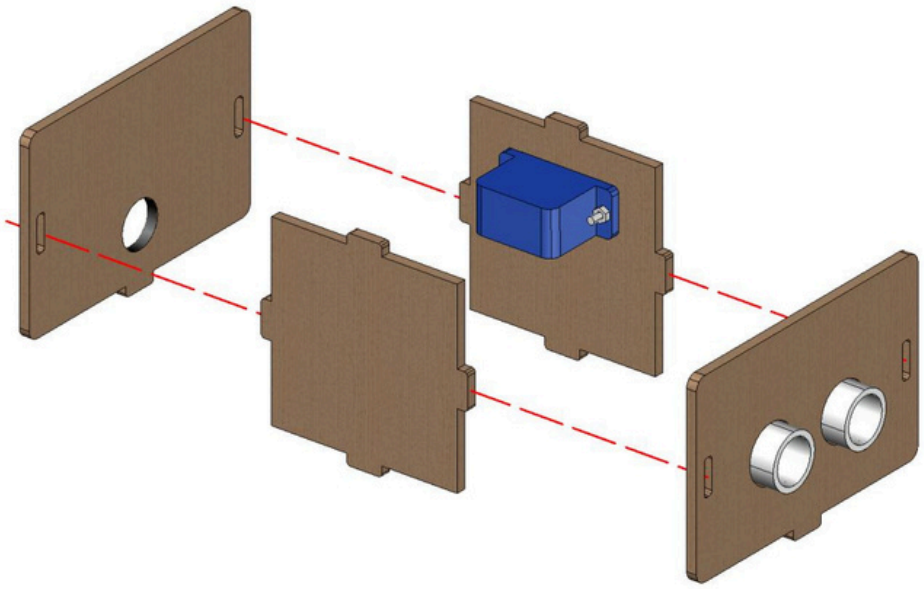
2. Seznam hardwaru

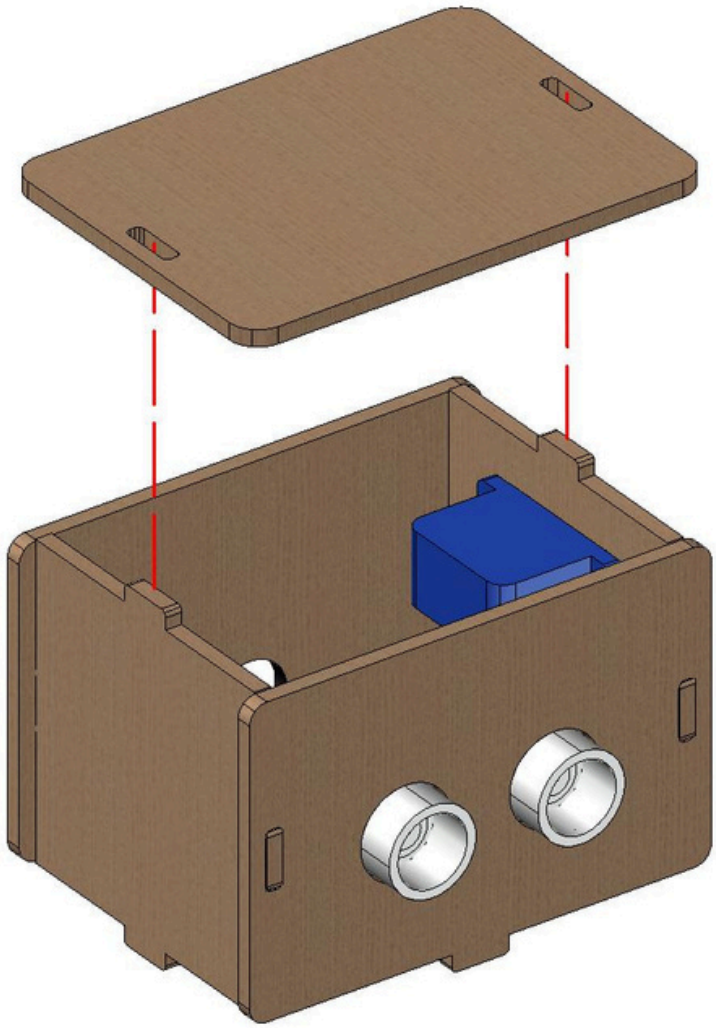
Obraz	Jméno	Množství
	Deska řadiče ESP32	1
	Ultrazvukový senzor	1
	Servo	1
	Nepájivé pole s 400 spojovacími body	1
	LED	1
	Rezistor 220Ω	1

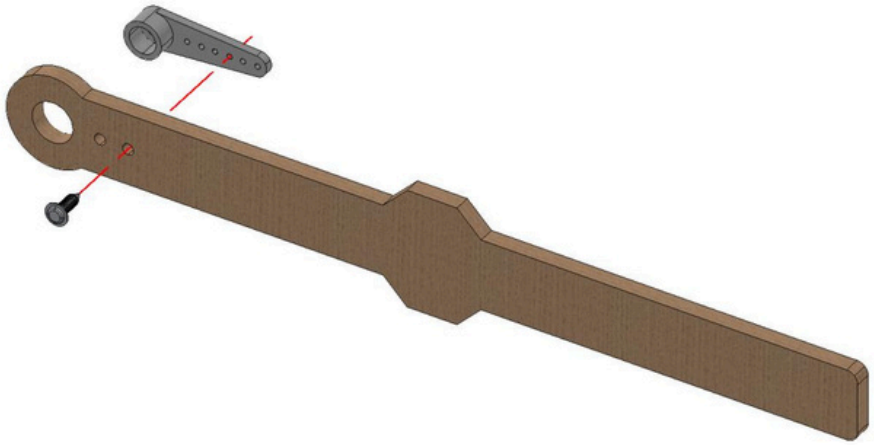
3. Sestavte model inteligentní brány (Pokud jste si nezakoupili stavební materiál, přeskočte kroky montáže a pokračujte přímo k připojení hardwaru a programování)

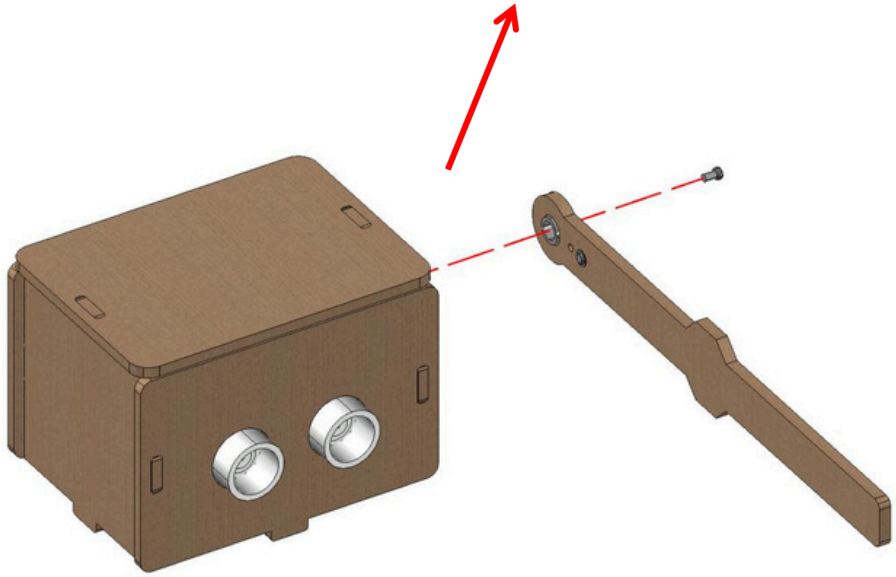
Krok 1 Instalace ultrazvukového modulu		
Seznam dílů	Ultrazvukový modul*1	Řada Dupont samec-samice*4
Ilustrace	 Připojte vodiče DuPont k pinům ultrazvukového modulu: červený vodič k VCC, modrý vodič k Trig, bílý vodič k Echo a černý vodič k GND. 	

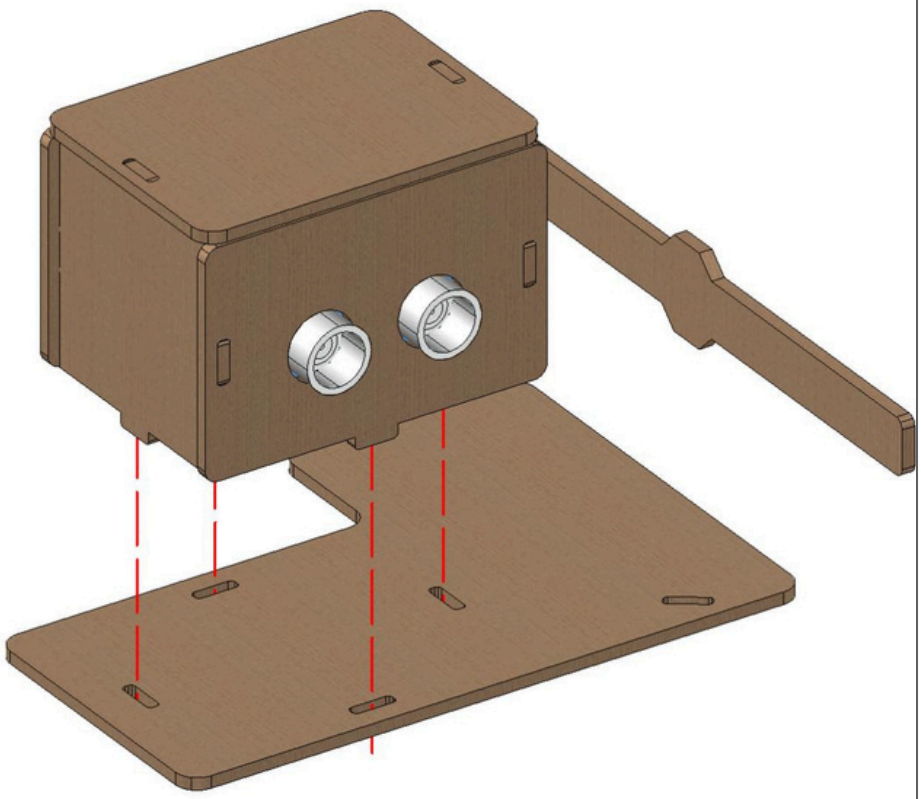
Krok 2 Instalace serva			
Seznam dílů	Servo *1	M2*14mm kulatá hlava Šrouby*2	Matice M2*2
Ilustrace	 		

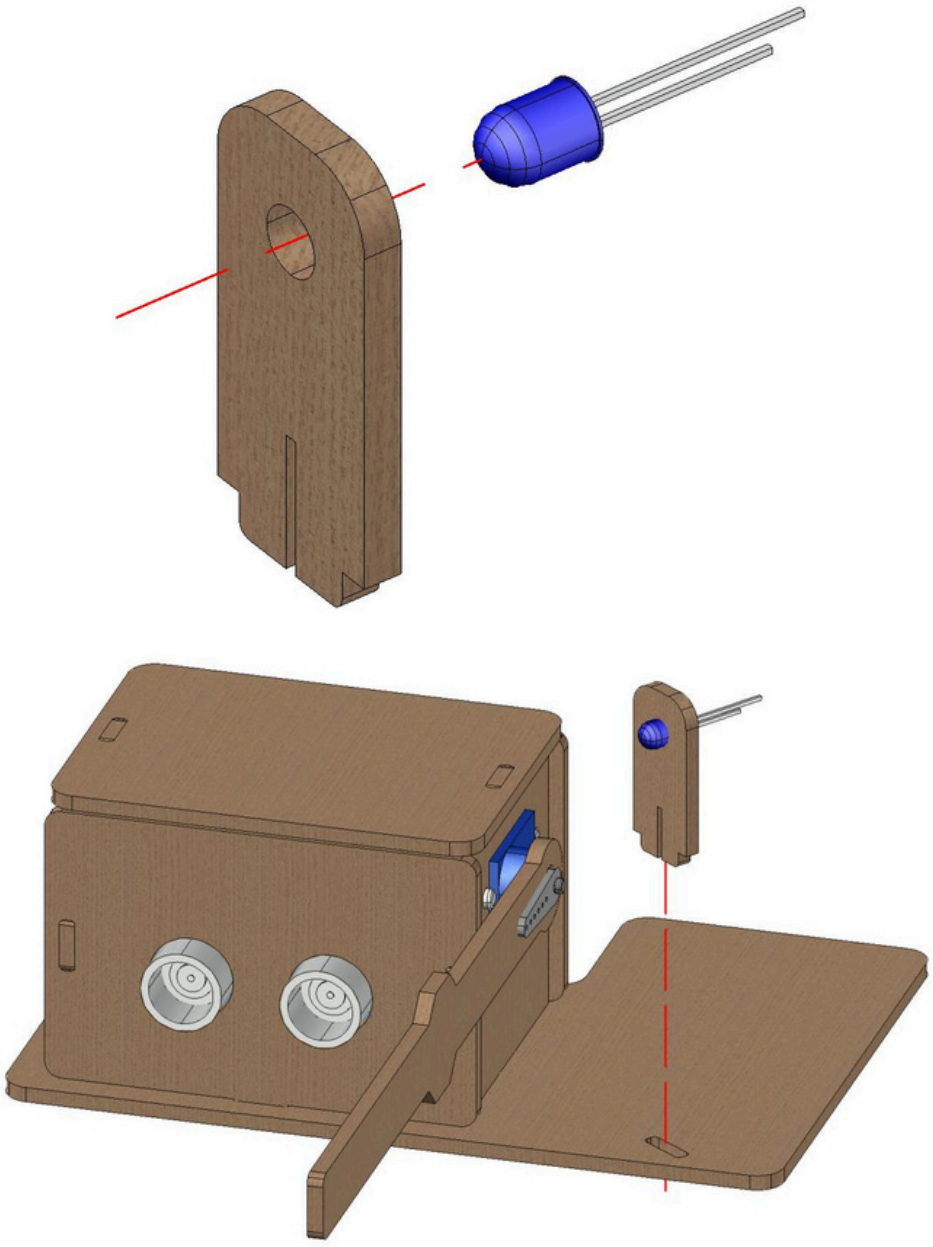
Krok 3 Instalace brány	
Ilustrace	

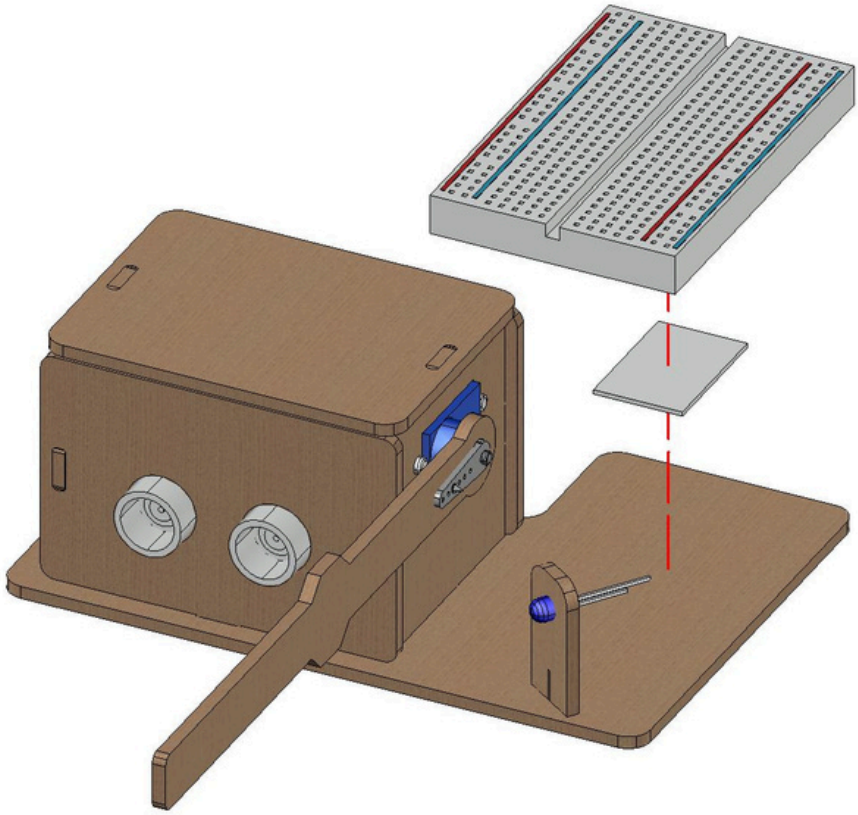
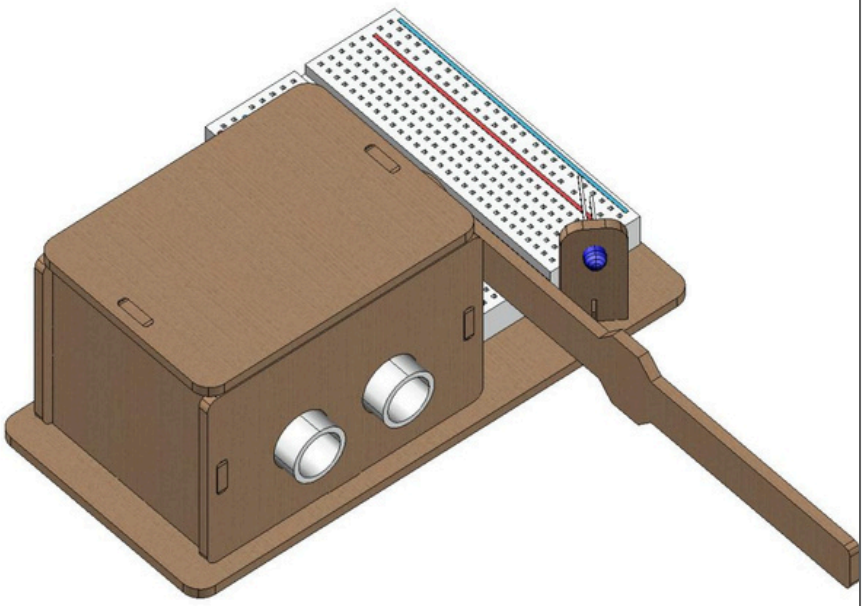
Krok 4 Instalace krytu brány	
Ilustrace	

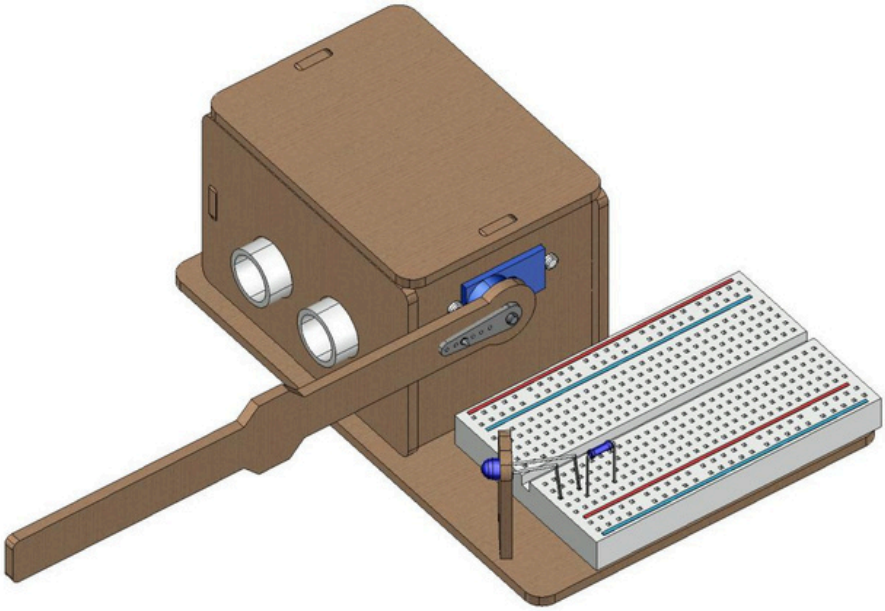
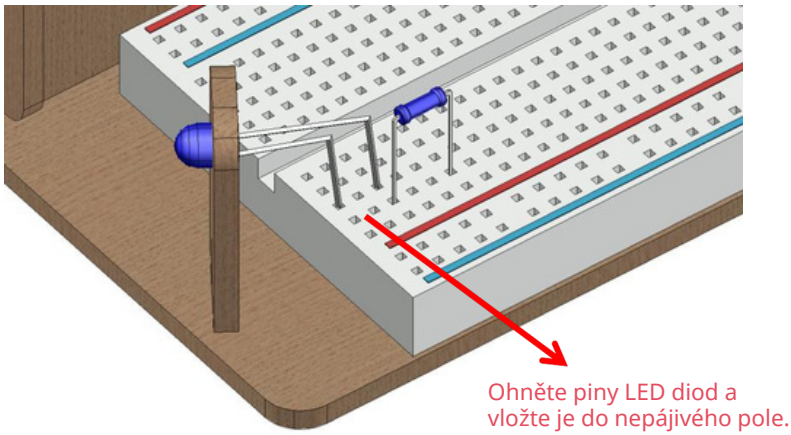
Krok 5 Nainstalujte volant		
Seznam dílů	M1,7*6 Samořezné Šroub*1	Řídicíkotouč s polovičními rameny*1
Ilustrace		

Krok 6. Instalace zvedací tyče	
Seznam dílů	Šrouby s kulatou hlavou M2,5*4*1
Ilustrace	Před zajištěním ramene výtahu nahrajte soubor „Servo_Initialize.ino“ do souboru „English\Arduino(Experienced Learner)\2.Code\lesson 35 Smart Barrier Gate\Servo_Initialize“, čímž otočíte servo o 90 stupňů. 

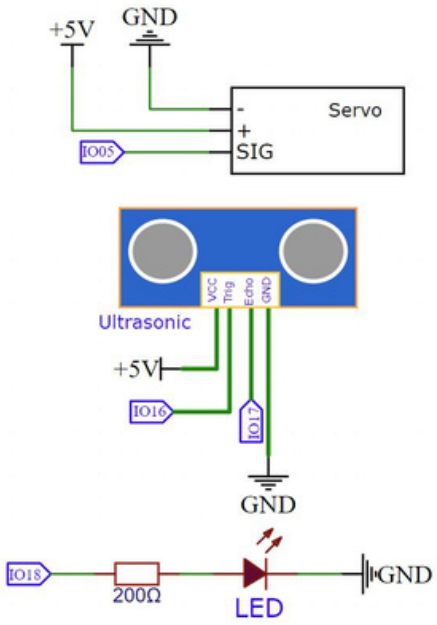
Krok 7 Instalace základny brány	
Ilustrace	

Krok 8 Instalace LED diody	
Seznam dílů	LED*1
Ilustrace	

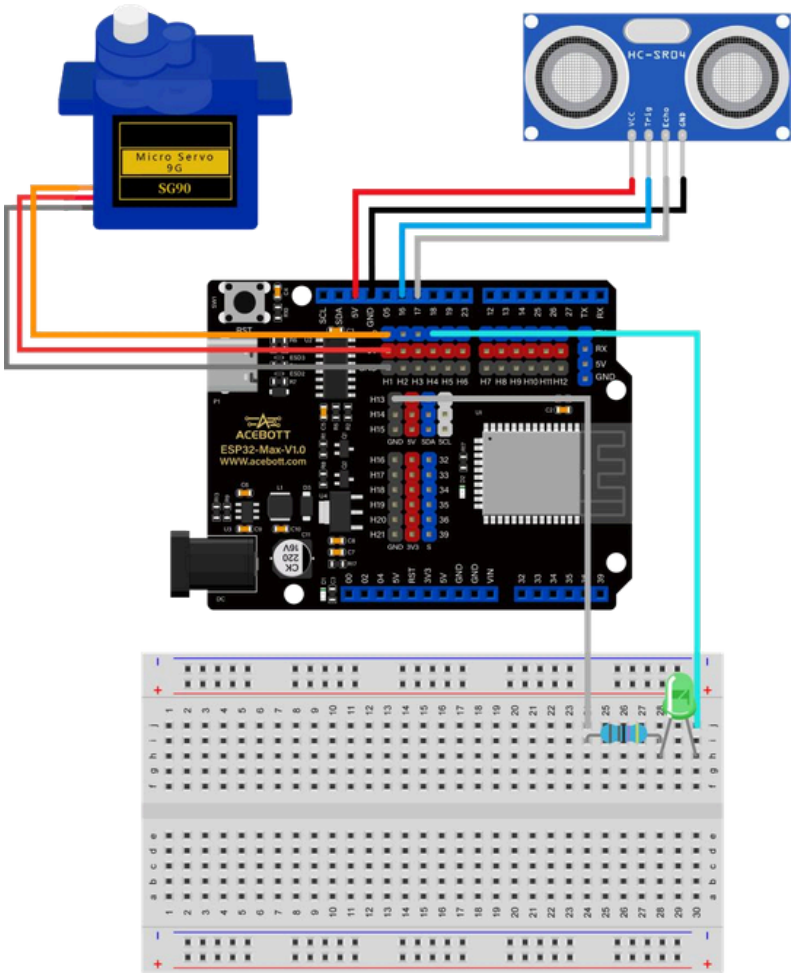
Krok 9. Instalace nepájivého pole		
Seznam dílů	Prkénko*1	Oboustranná páska*1
Ilustrace	 	

Krok 10 Instalace LED diody	
Seznam dílů	rezistor 220Ω
Ilustrace	 

4. Schéma zapojení



5. Připojení hardwaru



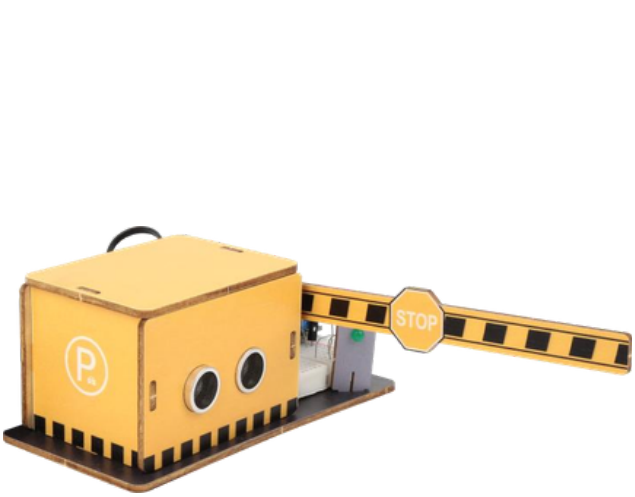
6. Referenční postupy

Otevřete "Smart_Barrier_Gate.ino „soubor v adresáři „English\Arduino(Experienced Learner)\2.Code\lesson 36 Smart Barrier Gate\Smart_Barrier_Gate“ nebo zkopírujte následující kód do Arduino IDE. Po výběru vývojové desky (ESP32 Dev Module) a příslušného portu podle dříve naučených kroků klikněte  nahrát program.

```
#include <ESP32Servo.h>

#define TRIG_PIN 16   // Ultrasonic sensor trigger pin
#define ECHO_PIN 17   // Ultrasonic sensor echo pin
#define LED PIN 18    // Indicator LED pin
#define SERVO_PIN 5   // Servo motor control pin

Servo servo;
int angle = 90;
int flag = 0;         // State flag: 0 = closed, 1 = opened

void setup() {
  Serial.begin(115200);           // Initialize serial monitor
  pinMode(TRIG PIN, OUTPUT);      // Set trigger pin as output
  pinMode(ECHO_PIN, INPUT);       // Set echo pin as input
  pinMode(LED_PIN, OUTPUT);       // Set LED pin as output
  servo.attach(SERVO_PIN);        // Attach servo to pin
  servo.write(angle);               // Default angle: 90° (closed
position)
}

void loop() {
  long distance = readDistance();   // Read distance from ultrasonic sensor
  Serial.println(distance);         // Print distance to serial monitor
  // If object (e.g., car) is detected within 14 cm and gate is closed
  if (distance > 0 && distance <= 14 && flag == 0) {
    digitalWrite(LED_PIN, HIGH);       // Turn on LED (indicates presence)
    for (angle = 90; angle > 0; angle--) {
      servo.write(angle);       // Gradually close the lid
      delay(5);                   // Small delay for smooth motion
    }
    flag = 1;                          // Update state to opened
  }
  // If object is gone (distance > 20 cm) and gate is opened
  else if (distance > 20 && flag == 1) {
    digitalWrite(LED_PIN, LOW);        // Turn off LED
      for (angle = 0; angle < 90; angle++) {
      servo.write(angle);       // Gradually open the lid
      delay(5);                   // Small delay for smooth motion
    }
```

```
    flag = 0;                          // Update state to closed
  }
  delay(500);  // Small delay for stability
}

// Function to read distance in centimeters from ultrasonic sensor
float readDistance() {
  digitalWrite(TRIG PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH); // Timeout after 30ms
  long distance = duration * 0.034 / 2;          // Convert time to distance
  delay(100);                                      // Small delay to
stabilize reading
  return distance;
}
```

Po úspěšném nahrání programu se rameno zvedne, když je před branou auto, a spustí se, když před ní žádné auto není.



Lekce 37 Oscilační ventilátor

Cíle kurzu

- Pochopte princip fungování oscilačního ventilátoru.
- Naučte se sestavit model oscilačního ventilátoru.

-Napište program pro implementaci funkcí řazení převodů a oscilace oscilačního ventilátoru.

Úvod do kurzu

Během horkého léta často používáme elektrické ventilátory k ochlazování. Na rozdíl od běžných ventilátorů se oscilační ventilátory mohou pohybovat doleva a doprava a rovnoměrně rozvádět chladný vzduch po celé místnosti. V moderních domácích spotřebičích je této schopnosti „automatického řízení“ dosaženo především motory nebo servopohony.

V této lekci se naučíme, jak pomocí servopohonů, motorů a programového řízení postavit jednoduchý „oscilační ventilátor“ a simulovat jeho přívod vzduchu a oscilační funkce!

Body znalostí

1. Jak funguje oscilační ventilátor?

Oscilační ventilátor se skládá ze dvou hlavních součástí: motoru, který pohání lopatky, aby se otáčely vysokou rychlostí a vytvářely tak nepřetržitý vánek; a oscilačního mechanismu, který způsobuje, že se hlava ventilátoru kývá doleva a doprava, čímž se vzduch rovnoměrně rozděluje po větší ploše. Tradiční oscilační ventilátory používají malý motor a převodovky k přeměně rotačního pohybu motoru na vratný pohyb hlavy ventilátoru. Moderní inteligentní oscilační ventilátory naopak často používají servopohony, které přijímají řídicí signály pro přesné nastavení úhlu kmitání, čímž se dosahuje flexibilní a programovatelné levé a pravé oscilace.

V reálném provozu motor otáčí lopatkami ventilátoru pro generování větru a oscilační mechanismus řídí hlavu ventilátoru tak, aby opakovaně oscilovala v přednastaveném rozsahu úhlů (například 45 stupňů doleva a doprava). Rychlost a amplitudu oscilace lze nastavit mechanickým provedením nebo programovým kódem. ESP32 přesně řídí servopohon nebo motor pomocí řídicích signálů, čímž dokončuje automatickou oscilační funkci oscilačního ventilátoru.

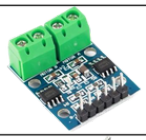
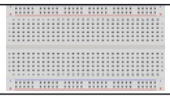
Experiment

1. Popis projektu

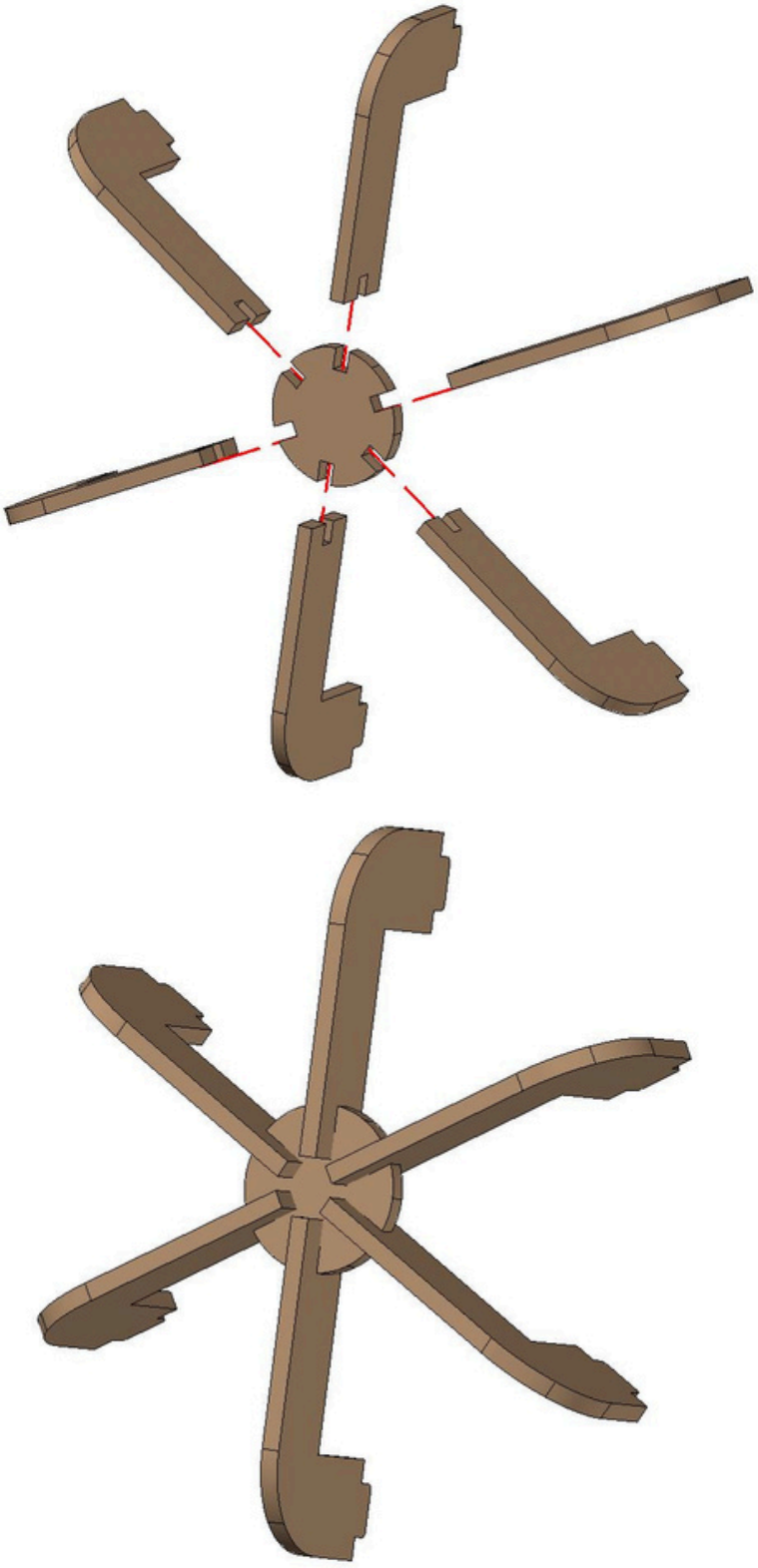
Tento program implementuje inteligentní systém oscilačního ventilátoru. Uživatelé mohou ventilátor zapínat a vypínat a ovládat rychlost větru (tři nastavení) jediným stisknutím tlačítka. Mohou také ovládat servo, které osciluje doleva a doprava a simuluje tak různé úhly proudění vzduchu. Systém kombinuje ovládání tlačítky, PWM regulaci rychlosti a řízení serva pro dosažení rychlosti větru.

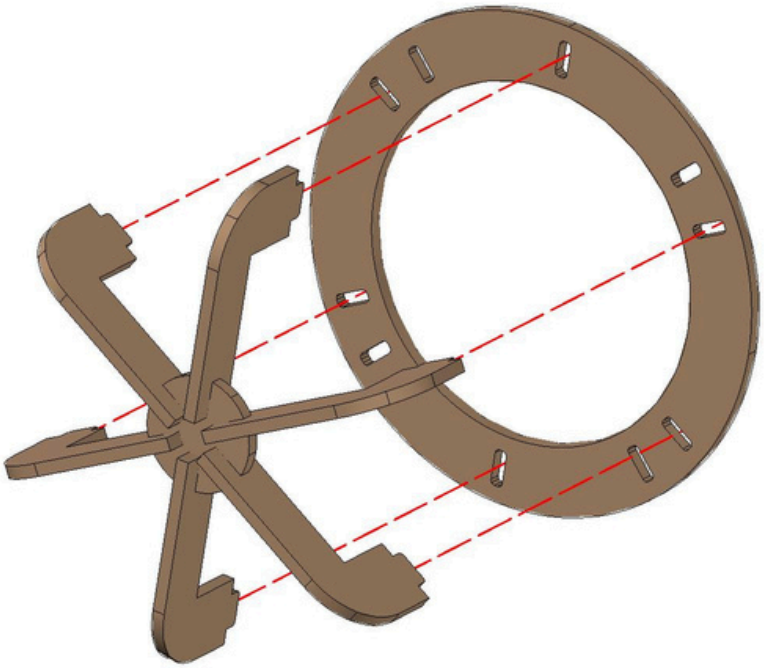
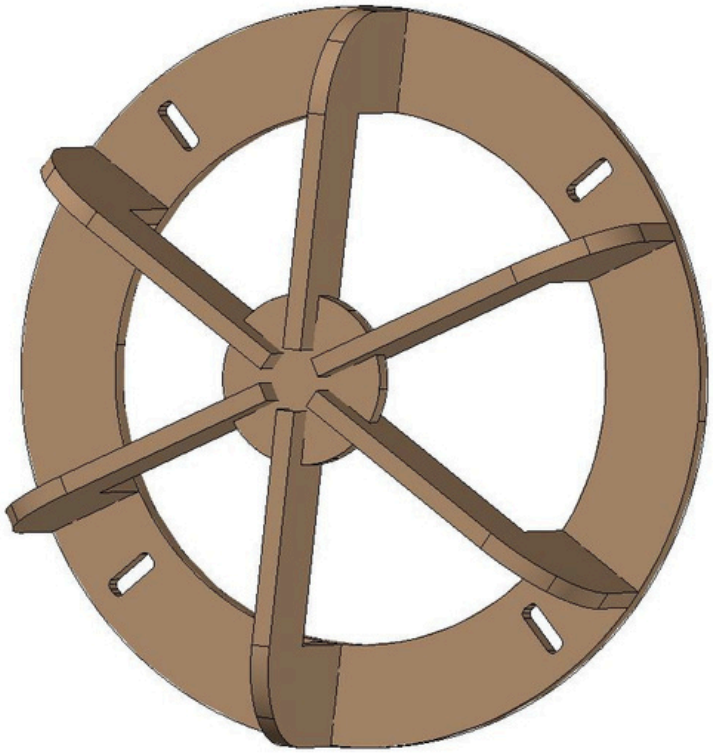
nastavení a oscilace, simulující uživatelský zážitek skutečného elektrického ventilátoru.

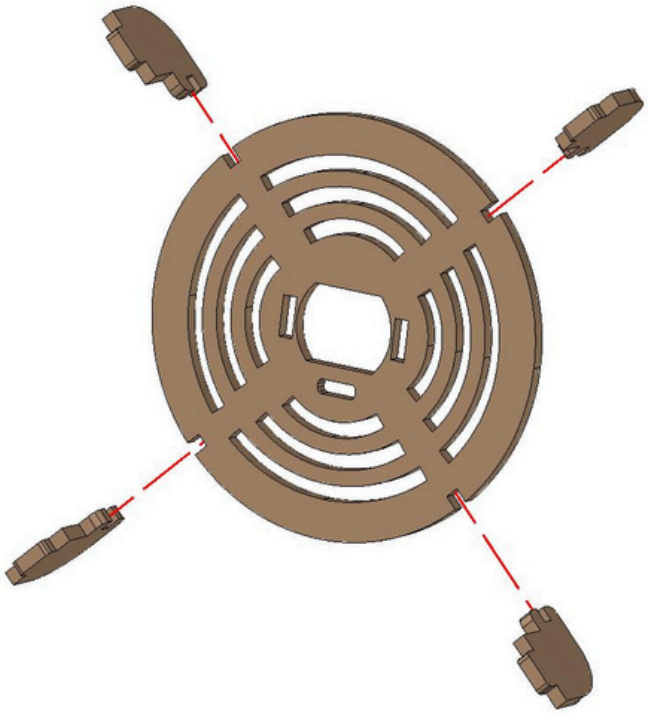
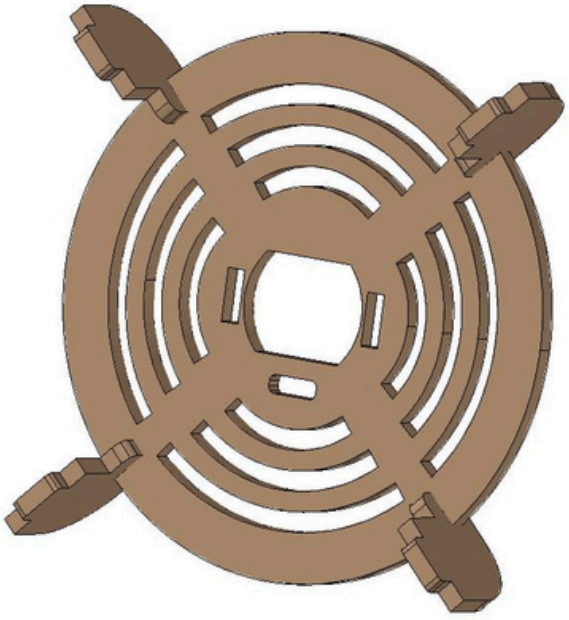
2. Seznam hardwaru

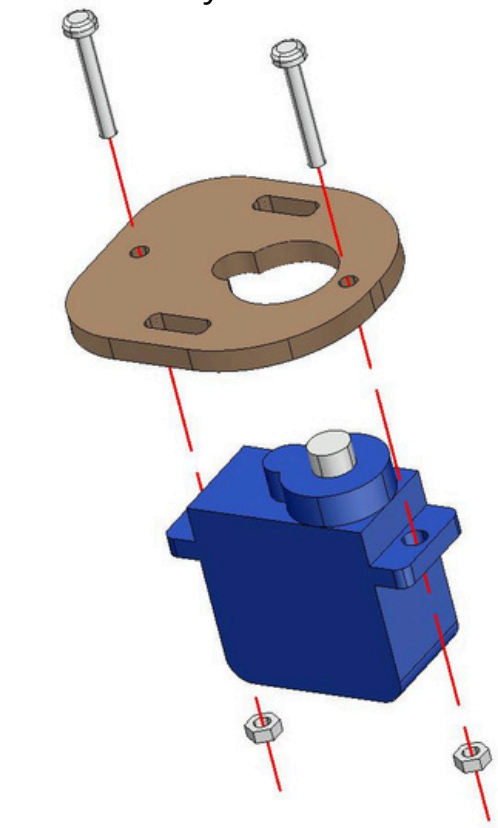
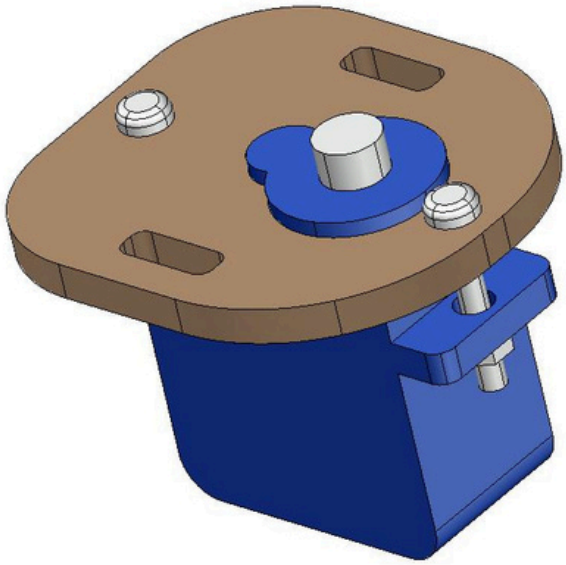
Obraz	Jméno	Množství
	Deska řadiče ESP32	1
	Ovladač stejnosměrného motoru	1
	Stejnosměrný motor	1
	Větrák	1
	Nepájivé pole s 400 spojovacími body	1
	Servo	1
	Tlačítko	3
	Rezistor 10 kΩ	3

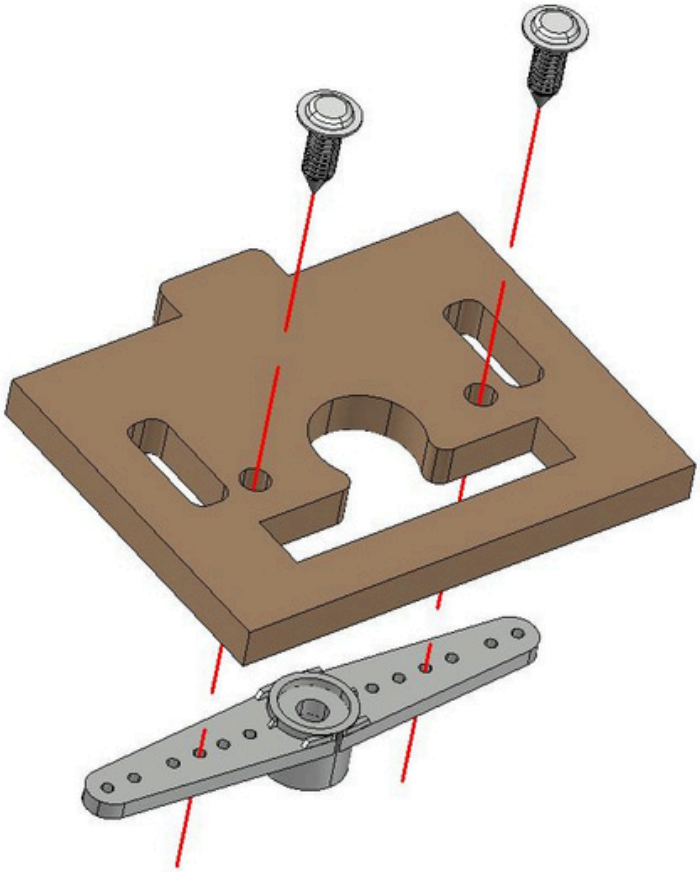
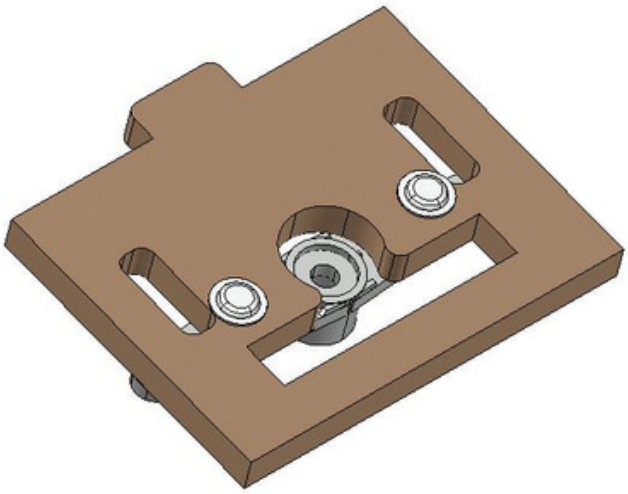
3. Sestavení oscilačního ventilátoru (Pokudjste si nezakoupili stavební materiál, přeskočte kroky montáže a pokračujte přímo k připojení hardwaru a učení programu)

Krok 1 Sestavte kryt předního ventilátoru	
Ilustrace	

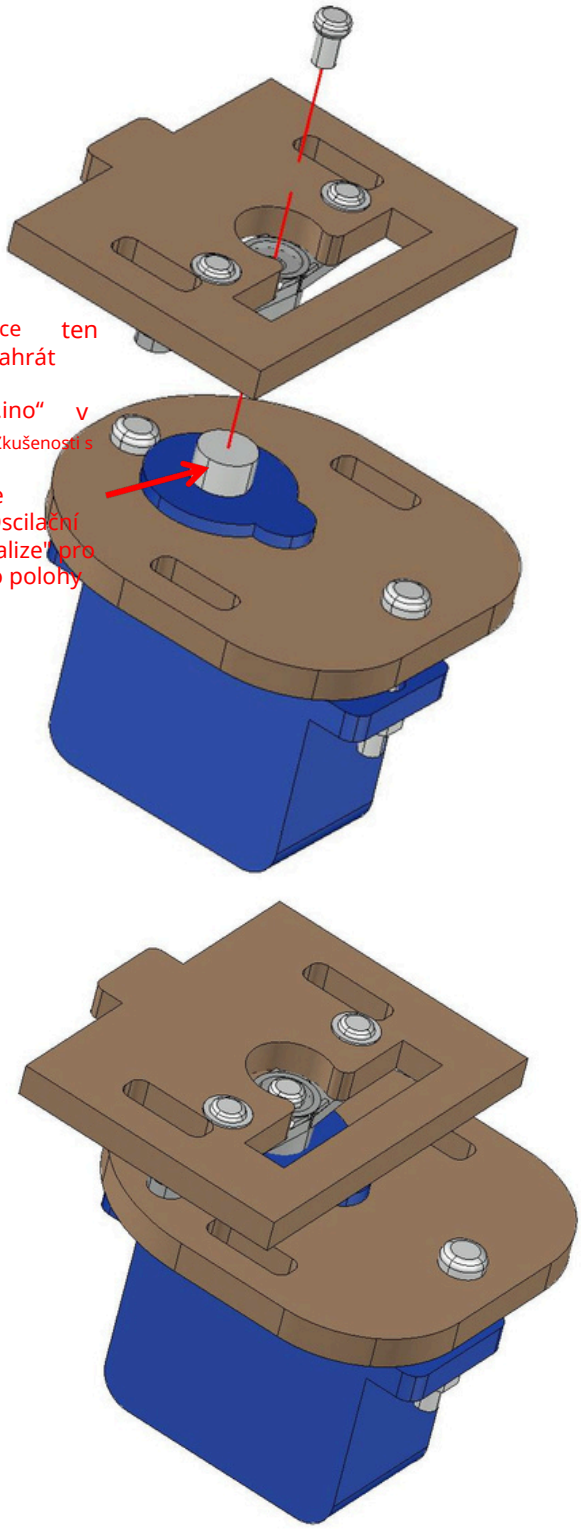
Krok 2. Instalace krytu předního ventilátoru	
Ilustrace	 

Krok 3. Sestavte kryt zadního ventilátoru	
Ilustrace	 

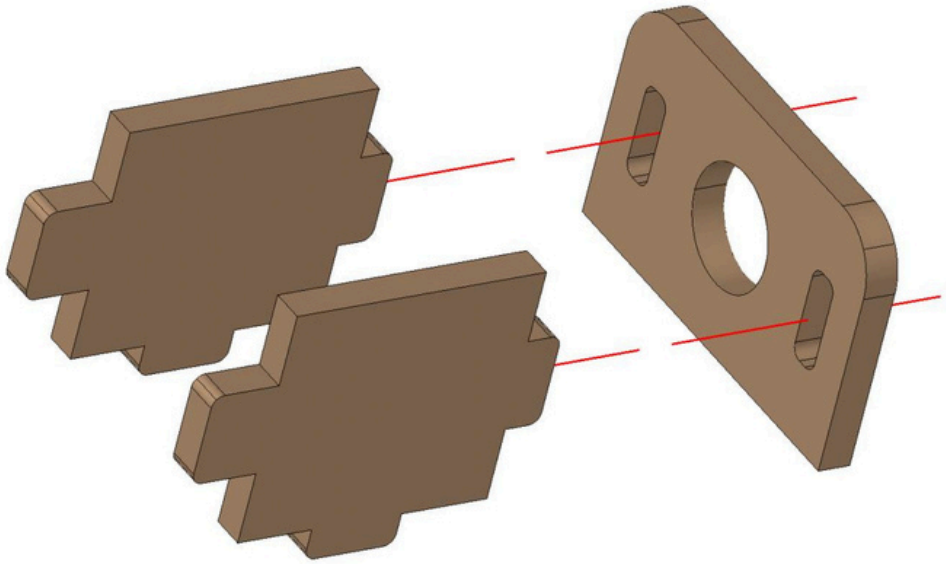
Krok 4 Instalace serva			
Seznam dílů	Servo *1	M2*14 kulatá hlava Šrouby*2	Matice M2*2
Ilustrace	 		

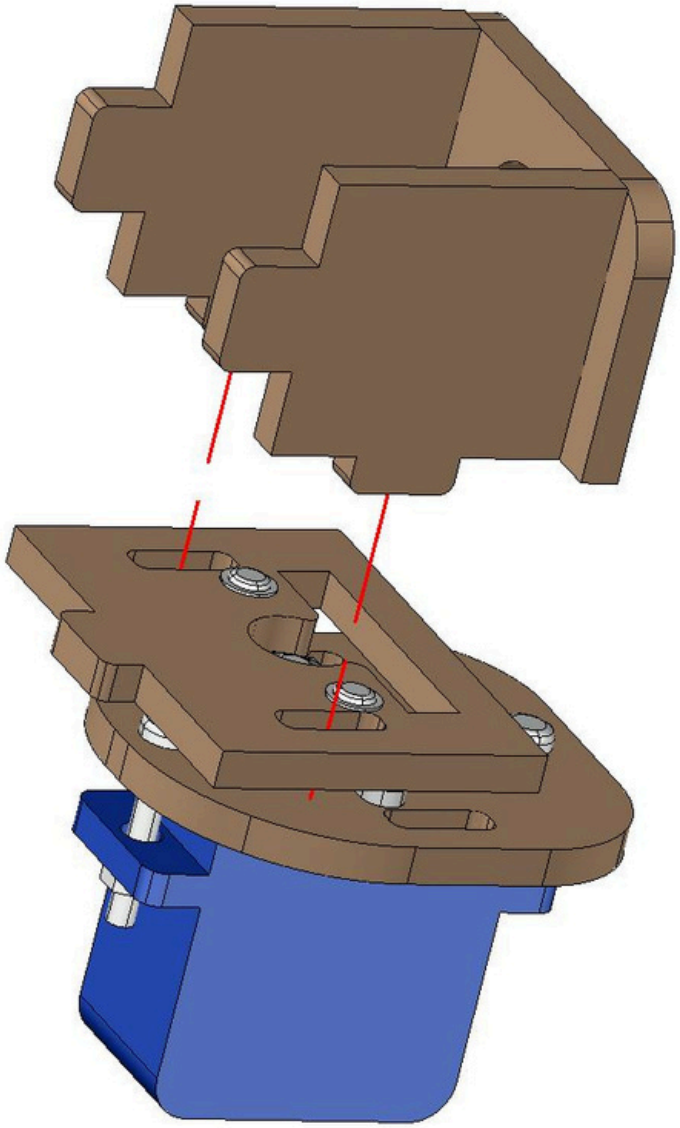
Krok 5. Nainstalujte volant		
Seznam dílů	Volant*1	2xvelké kulaté samořezné šrouby s plochou hlavou M1,7*6
Ilustrace	 	

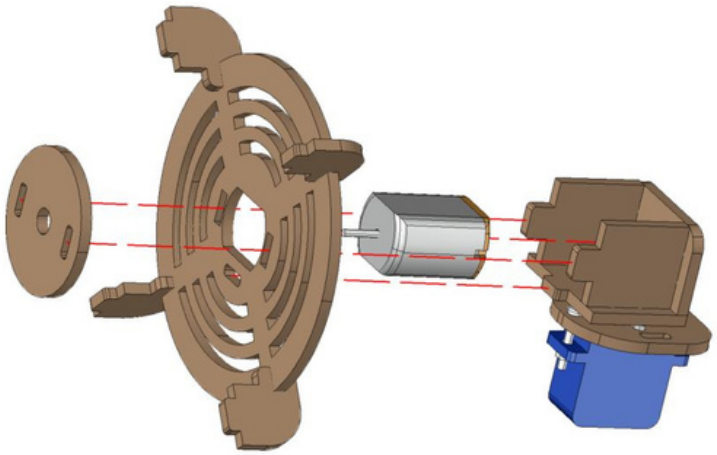
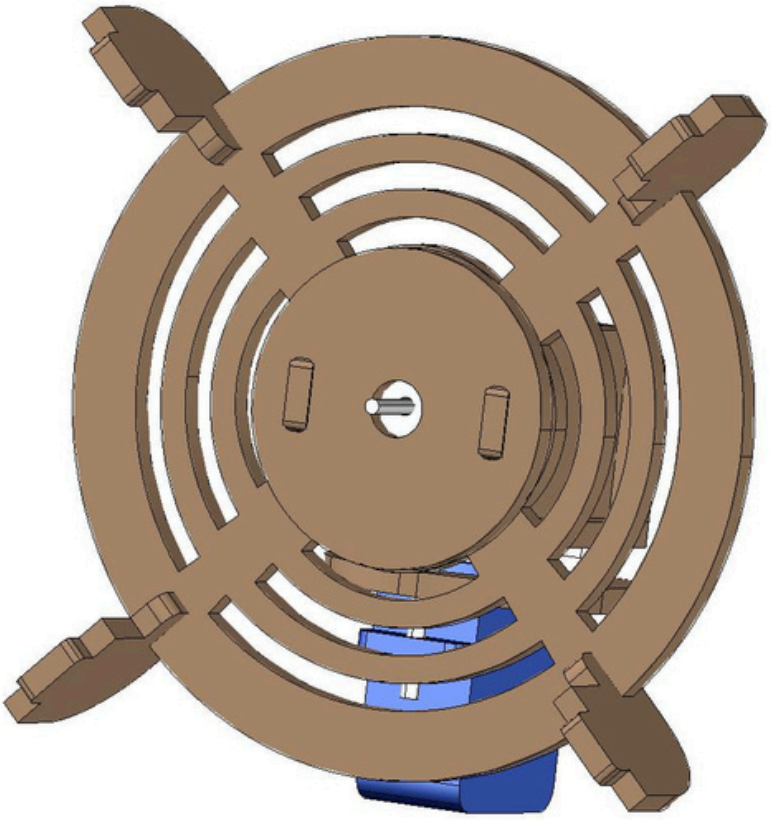
Krok 6. Zajištění volantu

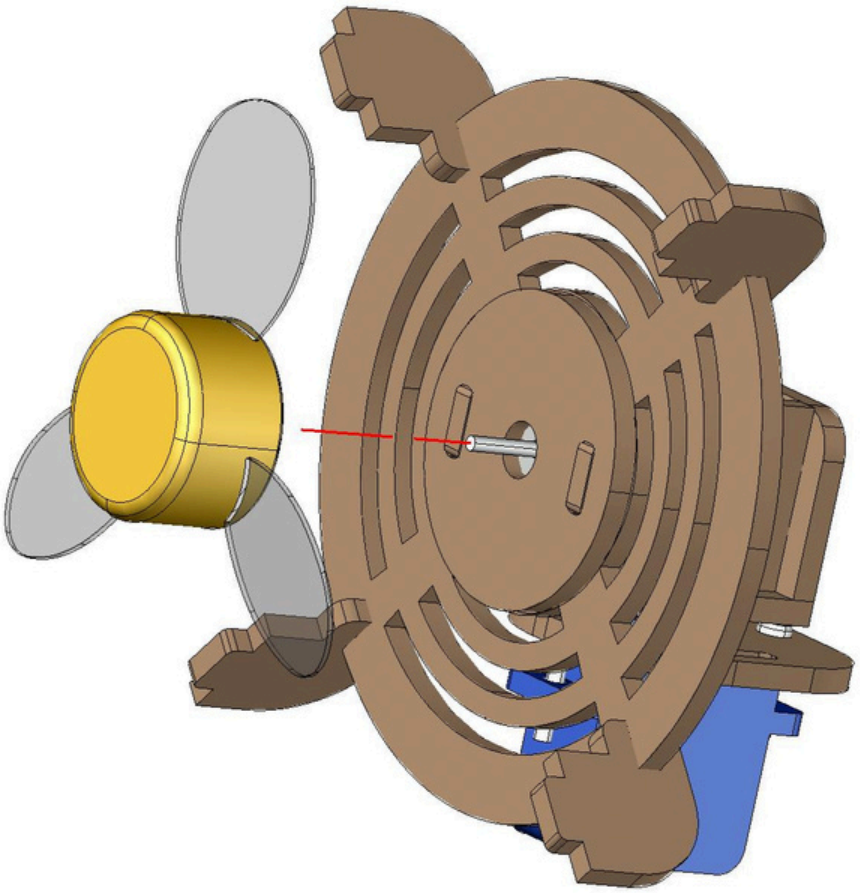
Seznam dílů	Šroub s kulatou hlavou M2,5*4*1
Ilustrace	

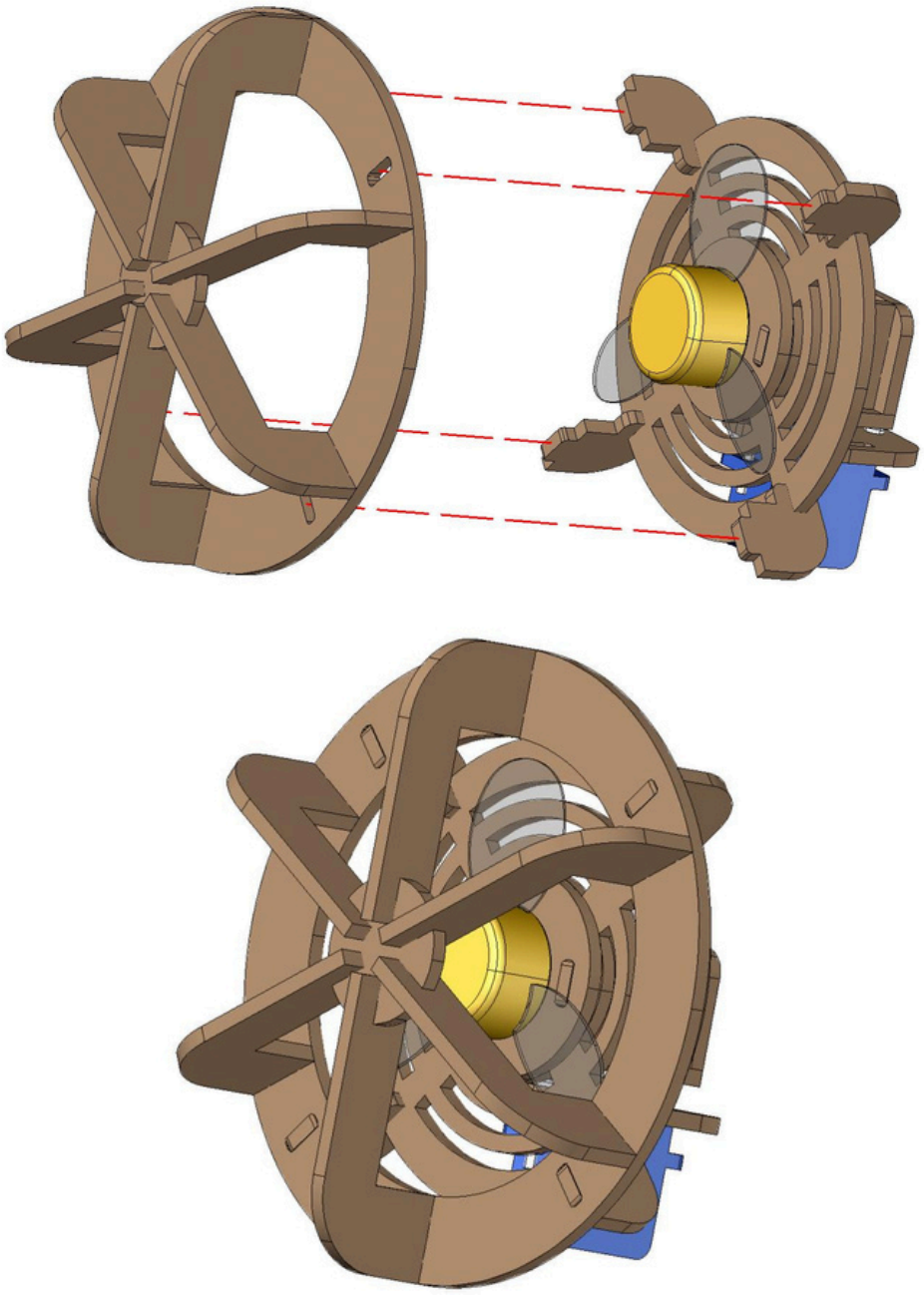
Před instalace ten řízení kolo,nahrát program „Servo_Initialize.ino“ v "Angličtina\Arduino(Zkušenosti s Žák)\2.Kód\lekce 36 Oscilační "Fan\Servo_Initialize" pro otočení serva do polohy 90 stupňů.

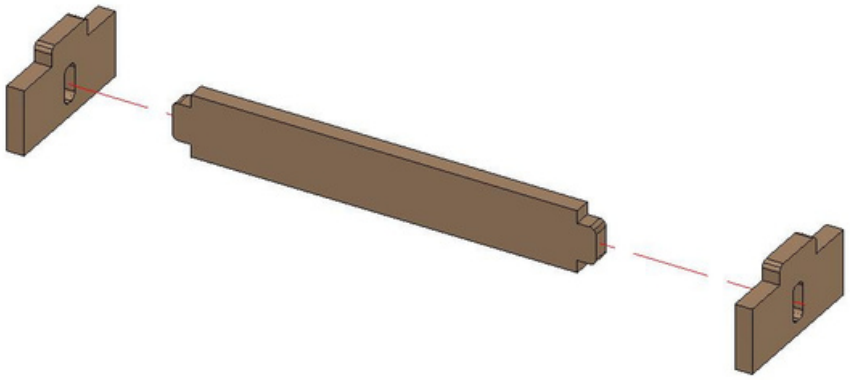
Krok 7. Sestavte držák motoru	
Ilustrace	

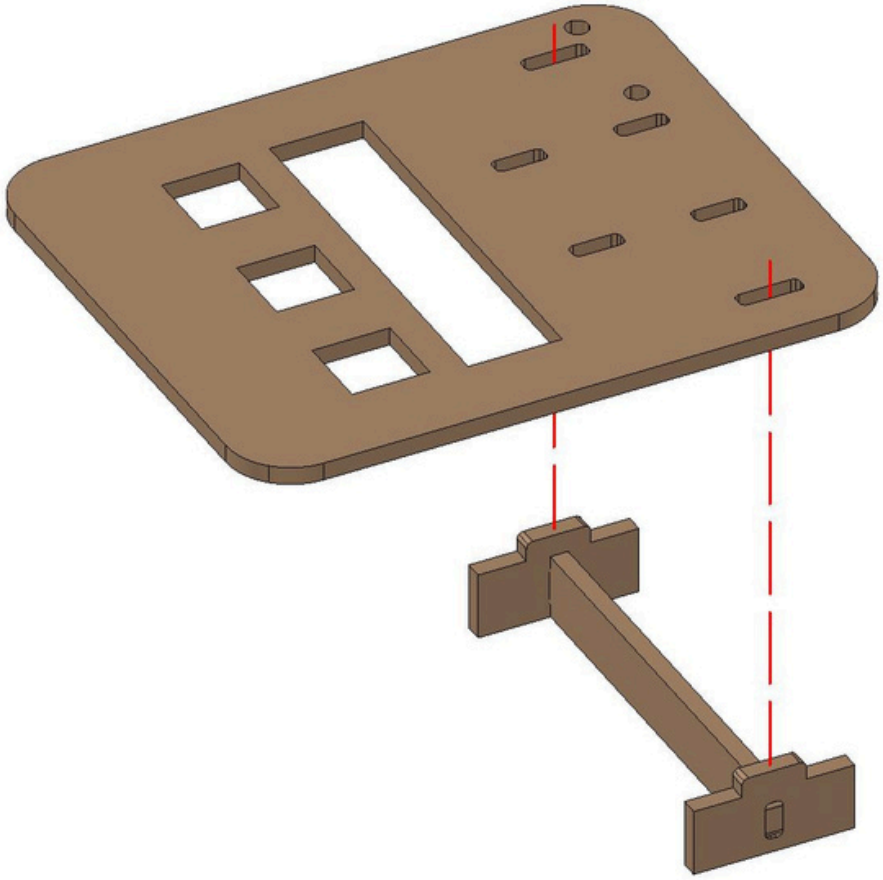
Krok 8. Instalace držáku motoru	
Ilustrace	

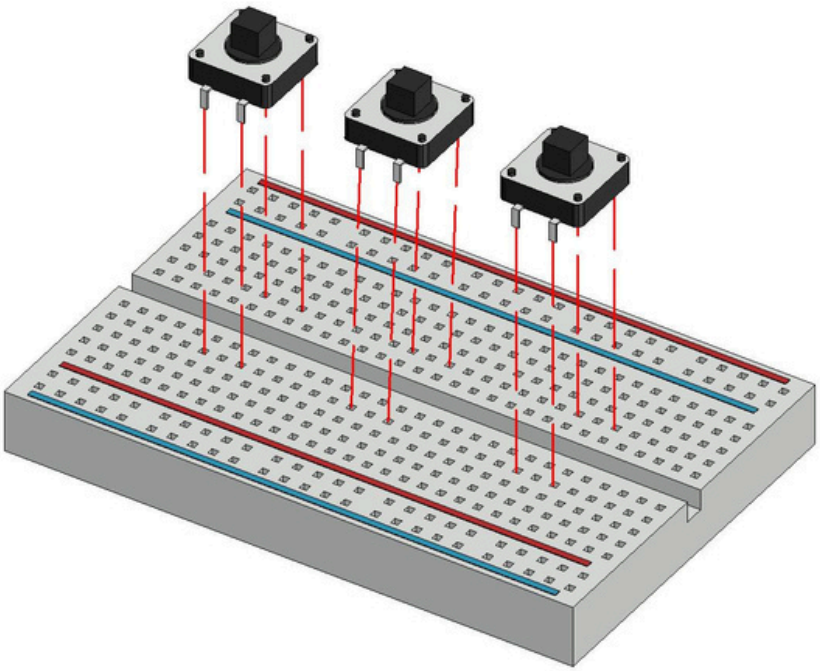
Krok 9 Instalace motoru a zadního krytu	
Seznam dílů	Motor*1
Ilustrace	 

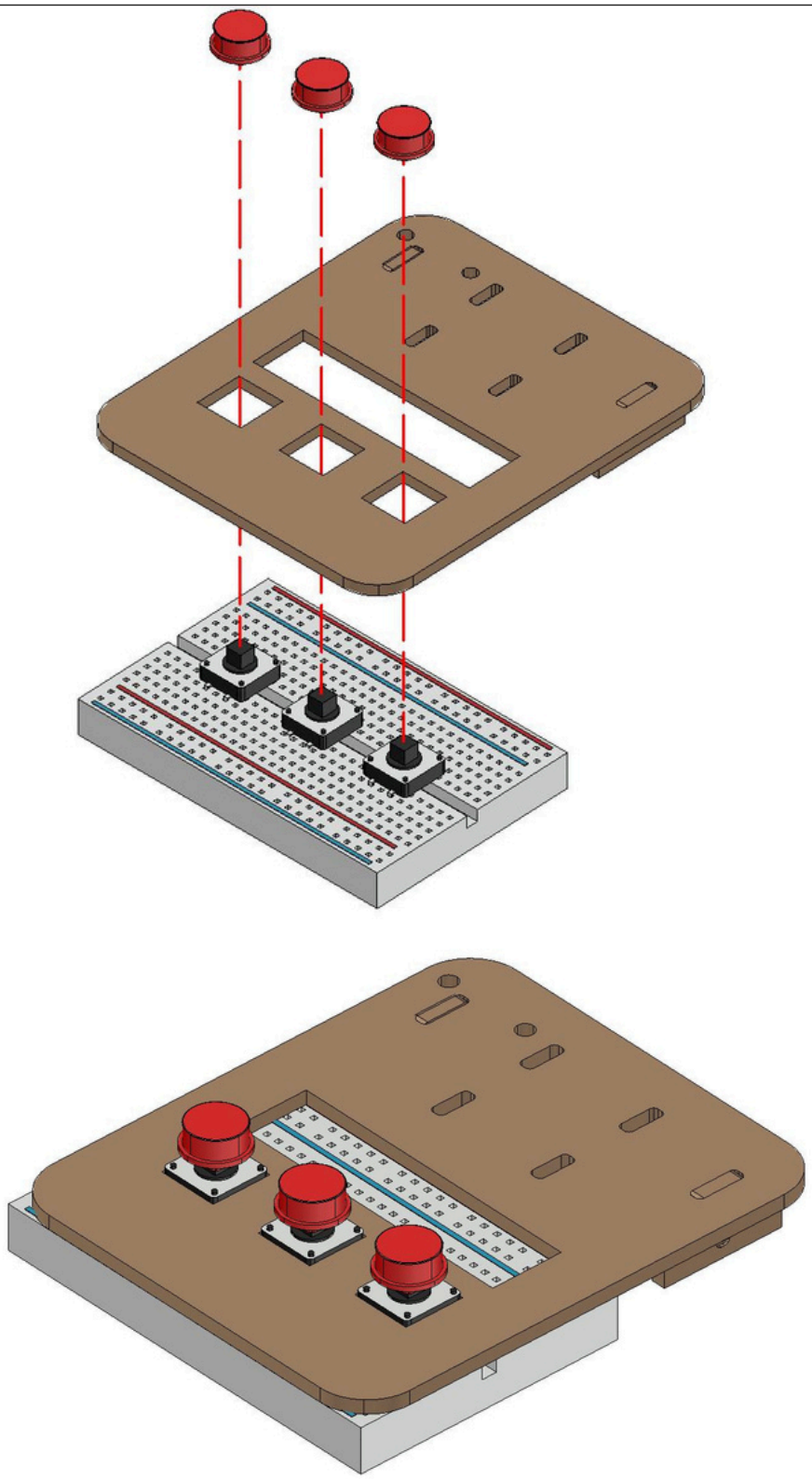
Krok 10 Nainstalujte lopatky ventilátoru	
Seznam dílů	Lopatka ventilátoru*1
Ilustrace	

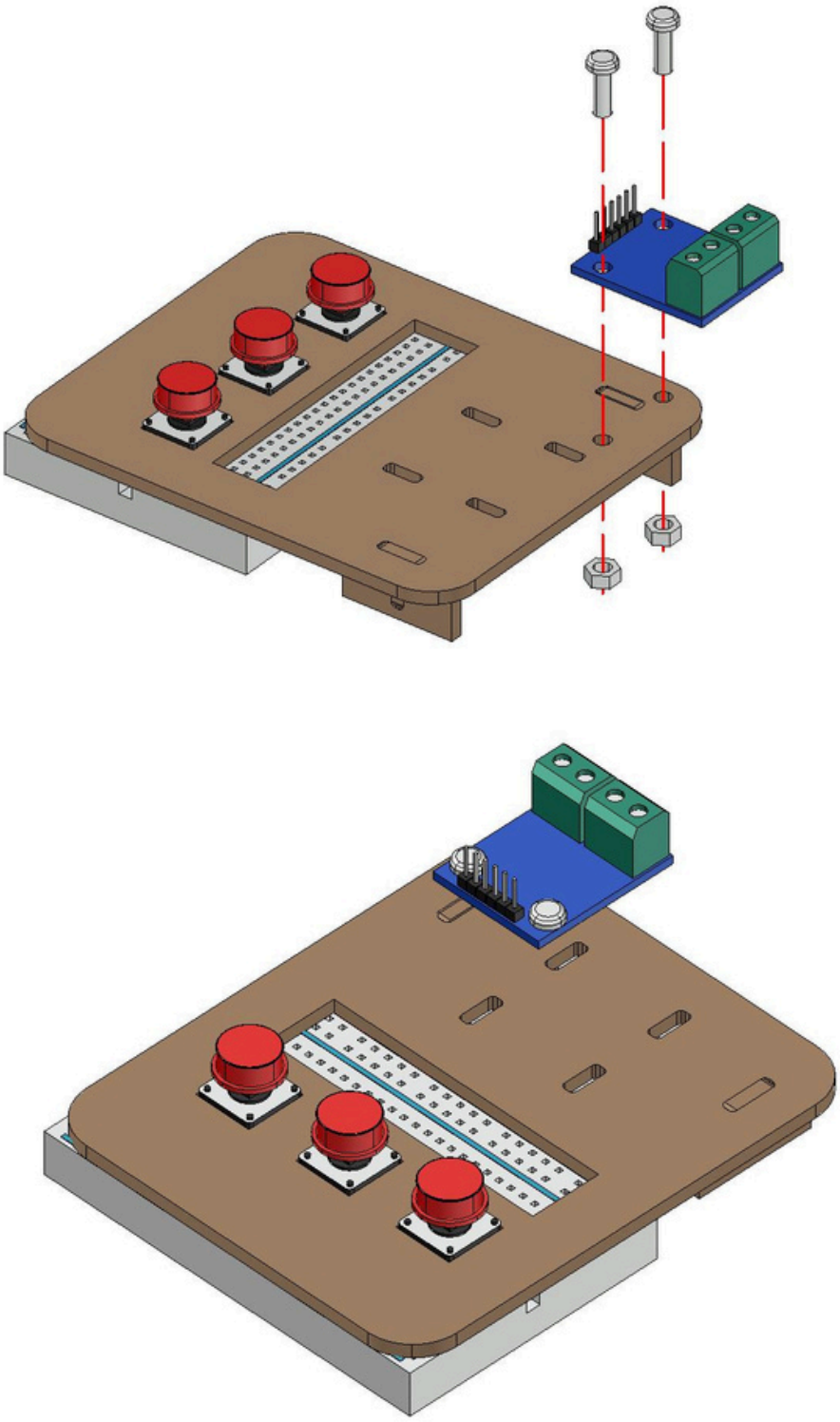
Krok 11 Nainstalujte přední kryt	
Ilustrace	

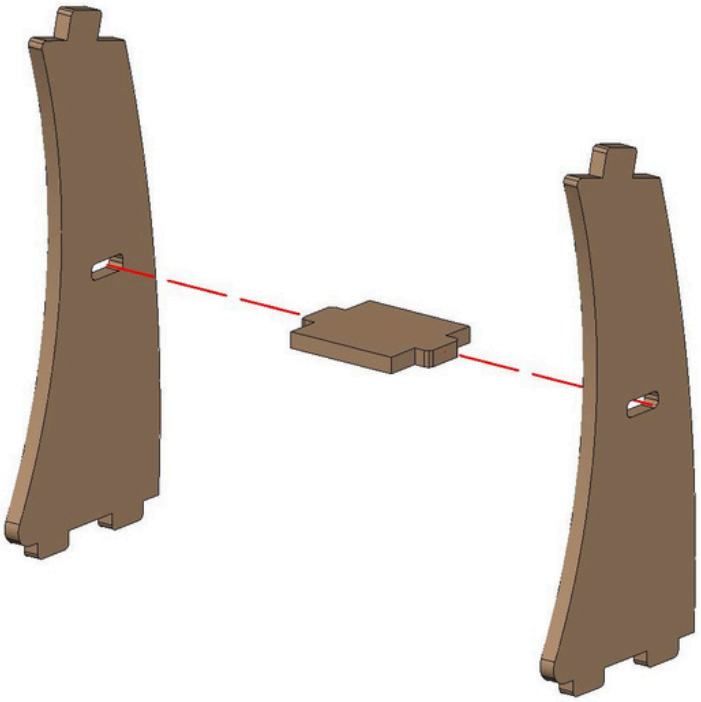
Krok 12 Instalace podpěrné konzoly základny ventilátoru	
Ilustrace	

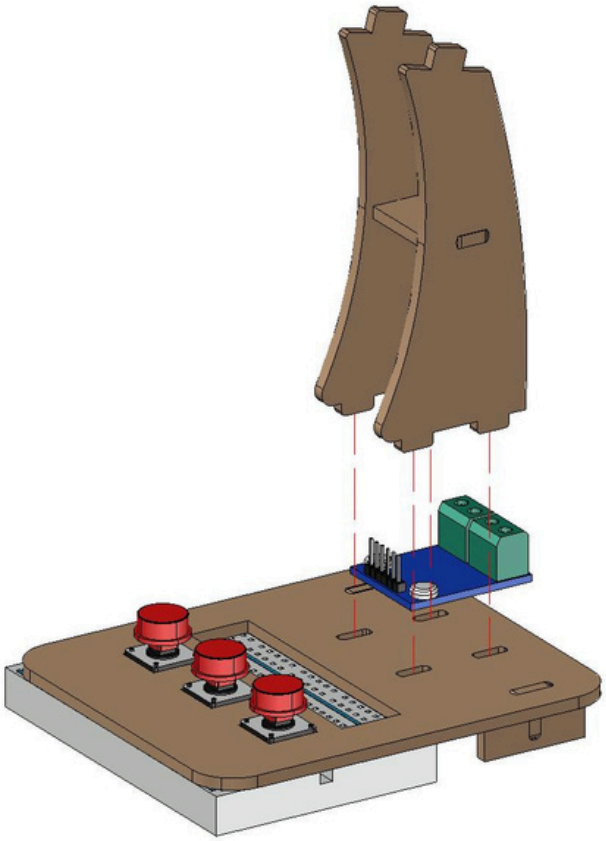
Krok 13 Sestavte základnu ventilátoru	
Ilustrace	

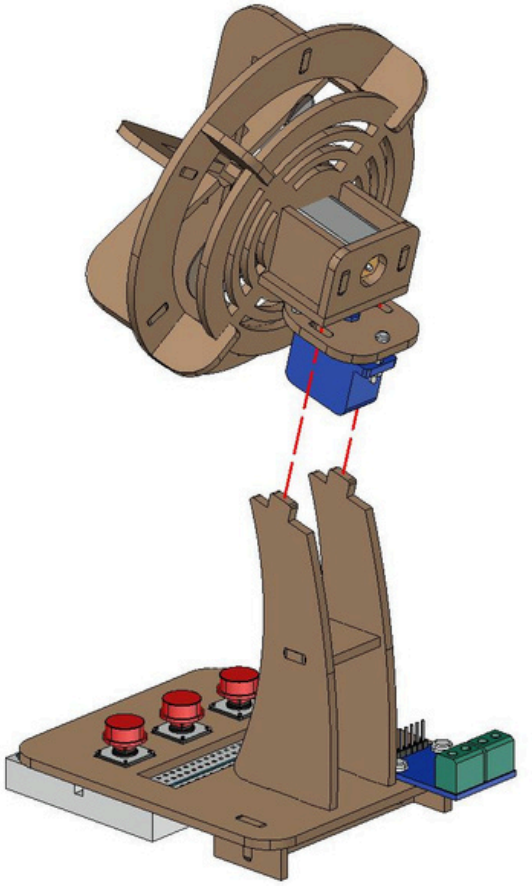
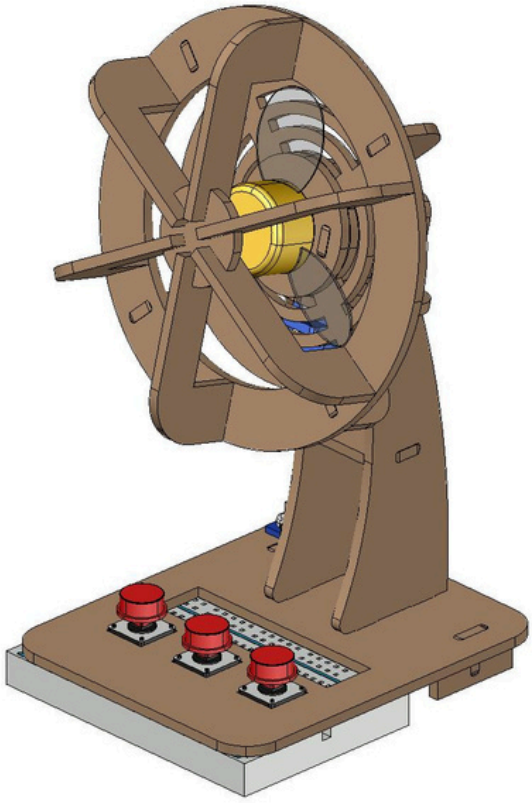
Krok 14 Instalace tlačítka		
Seznam dílů	Tlačítka*3	Prkénko*1
Ilustrace		

Krok 15. Instalace kláves	
Seznam dílů	Klávesy*3
Ilustrace	

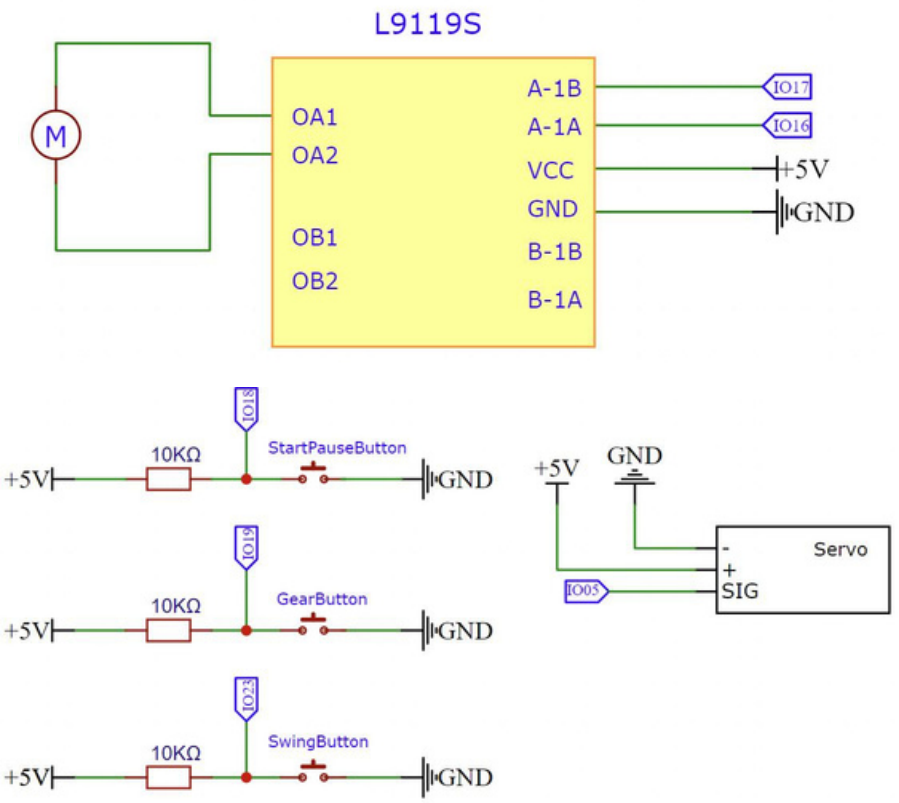
Krok 16 Instalace desky ovladače motoru			
Seznam dílů	M3*10mm kulatá hlava Šroub*2	Matice M3*2	Ovladač motoru Deska*1
Ilustrace			

Krok 17 Sestavte podpůrné sloupky	
Ilustrace	

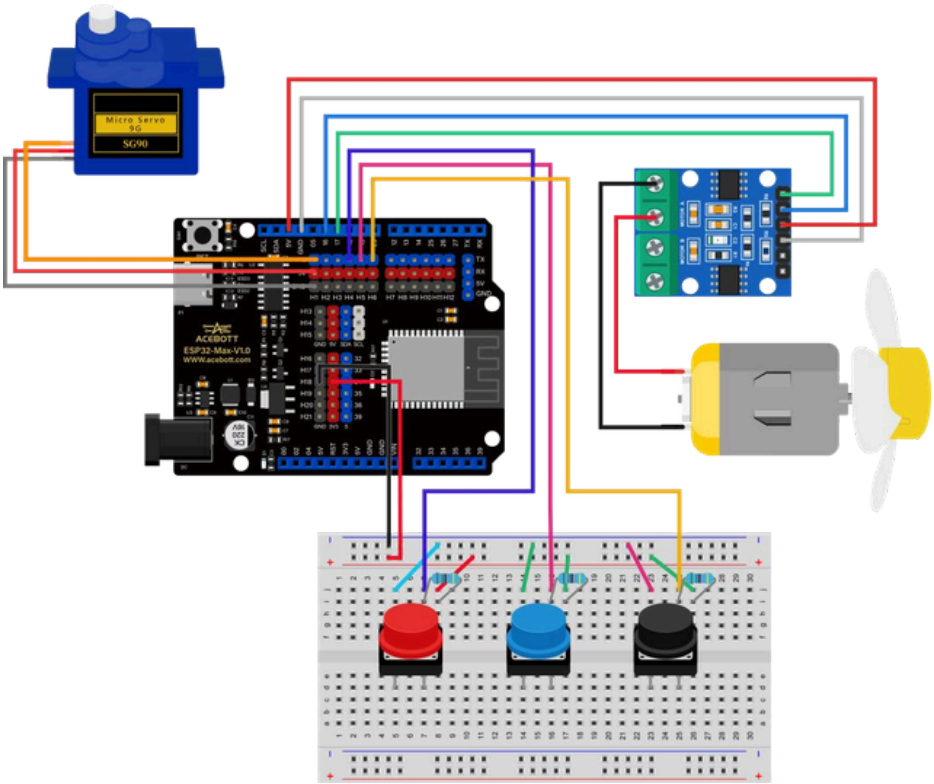
Krok 18 Instalace podpůrných sloupků	
Ilustrace	

Krok 19 Zajistěte sběrnici ventilátoru	
Ilustrace	 

4. Schéma zapojení



5. Hardwarová připojení



6. Referenční program

Otevřete "Oscilating_Fan.ino „soubor v adresáři „English\Arduino(Experienced Learner)\2.Code\lesson 376 Oscilating Fan\Oscillating_Fan“ nebo zkopírujte následující kód do Arduino IDE. Po výběru vývojové desky (ESP32 Dev Module) a příslušného portu podle pokynů dříve naučené kroky, klikněte  nahrát program.

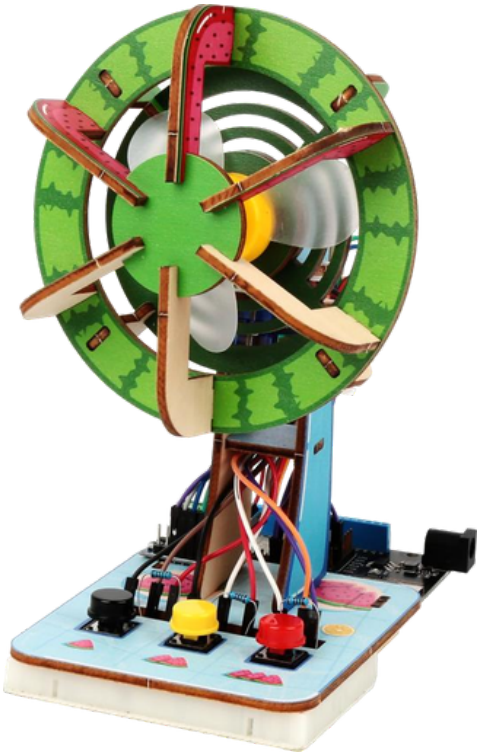
```
#include <ESP32Servo.h>

const int servoPin = 5; // Servo connected to pin 5 on ESP32
const int fanPin = 16; // Fan control pin
const int unusedPin = 17; // Unused pin, kept LOW
const int buttonPin1 = 18;
const int buttonPin2 = 19;
const int buttonPin3 = 23;

Servo myServo; // Create a servo object
bool isMoving = false; // Whether the servo is currently moving
float angle = 90;
int dutyCycle = 0;
int currentFanLevel = 0;
int isWaving = 0;
int is_speedcontrol = 0;
unsigned long lastDebounceTime = 0; // Last debounce time
const unsigned long debounceDelay = 50; // Debounce delay in milliseconds
int reading_1;

int buttonState1 = 0; // Current state of button 1
int lastButtonState1 = 0; // Previous state of button 1
int buttonToggleState1 = 0; // Toggle state for button 1 (0 or 1)

int buttonState2 = 0; // Current state of button 2
int lastButtonState2 = 0; // Previous state of button 2
int buttonToggleState2 = 0; // Toggle state for button 2 (unused in logic)

int buttonState3 = 0; // Current state of button 3
int lastButtonState3 = 0; // Previous state of button 3
int buttonToggleState3 = 0; // Toggle state for button 3 (0 or 1)

void setup() {
Serial.begin(115200); // Initialize serial communication for debugging
pinMode(buttonPin1, INPUT); // Set button 1 as input
pinMode(buttonPin2, INPUT); // Set button 2 as input
pinMode(buttonPin3, INPUT); // Set button 3 as input
pinMode(fanPin, OUTPUT);
```

```
pinMode(unusedPin, OUTPUT);

// Set up PWM for fan using ledcWrite
ledcSetup(0, 20000, 8); // Channel 0, 20kHz frequency, 8-bit resolution
ledcAttachPin(fanPin, 0); // Attach fan pin to channel 0

// Configure servo library
ESP32PWM::allocateTimer(1); // Allocate a timer for PWM use
myServo.setPeriodHertz(50); // Standard 50Hz for servo
myServo.attach(servoPin, 500, 2500);  // Attach servo to pin and define
pulse width range

digitalWrite(unusedPin, LOW); // Ensure unused pin stays LOW
myServo.write(90); // Set initial servo angle to 90°
delay(1000); // Wait for servo to stabilize
}

void loop() {
button(); // Check button states

// If servo is set to move, start waving
if (isMoving) {
startWaving();
}
}

void button() {
// Read button 1 state
reading_1 = digitalRead(buttonPin1);
if (reading_1 != lastButtonState1) {
lastDebounceTime = millis(); // Timestamp for debounce
}

if ((millis() - lastDebounceTime) > debounceDelay) {
if (reading_1 != buttonState1) {
buttonState1 = reading_1;
if (buttonState1 == HIGH) {
if (buttonToggleState1 == 0) {
buttonToggleState1 = 1;
} else if (buttonToggleState1 == 1) {
buttonToggleState1 = 0;
}
}

// Toggle logic for button 1
if (buttonToggleState1 == 1) {
Serial.println("000");
isWaving = 1;
is_speedcontrol = 1;
```

```
currentFanLevel = 3;
speed control(currentFanLevel);
} else if (buttonToggleState1 == 0) {
Serial.println("111");
currentFanLevel = 0;
speed_control(currentFanLevel);
isWaving = 0;
is_speedcontrol = 0;
isMoving = false; // Stop servo movement
}
}
}
lastButtonState1 = reading_1;

// Read button 2 state
buttonState2 = digitalRead(buttonPin2);
if (buttonState2 == LOW && lastButtonState2 == HIGH) {  // On button press
currentFanLevel += 1;
if (currentFanLevel > 3) {
currentFanLevel = 1;
}
delay(50);  // Simple debounce
speed_control(currentFanLevel);
}
lastButtonState2 = buttonState2;

// Read button 3 state
buttonState3 = digitalRead(buttonPin3);
if (buttonState3 == LOW && lastButtonState3 == HIGH) {  // On button press
buttonToggleState3 = !buttonToggleState3;
if (buttonToggleState3 == 1 && isWaving == 1) {
isMoving = true;  // Start waving
} else if (buttonToggleState3 == 0) {
isMoving = false; // Stop waving
}
delay(50); // Simple debounce
}
lastButtonState3 = buttonState3;
}

void startWaving() {
// Simple back-and-forth waving motion
static bool goingUp = true;  // Controls direction

if (goingUp) {
angle += 1;
if (angle >= 170) {
goingUp = false;
}
```

```
} else {
angle -= 1;
if (angle <= 10) {
goingUp = true;
}
}

myServo.write(angle);
delay(20);  // Adjust delay to control speed of waving
}

void speed_control(int command) {
if (is_speedcontrol == 1) {
switch (command) {
case 0:
dutyCycle = 0;
Serial.println("Fan set to 0% speed");
break;
case 1:
dutyCycle = 200;
Serial.println("Fan set to 50% speed");
break;
case 2:
dutyCycle = 220;
Serial.println("Fan set to 75% speed");
break;
case 3:
dutyCycle = 235;
Serial.println("Fan set to 100% speed");
break;
default:
Serial.println("Invalid command. Use 1, 2, or 3.");
return;
}
ledcWrite(0, dutyCycle);  // Apply PWM duty cycle to fan
}
}
```

Po úspěšném nahrání programu stiskněte černé tlačítko pro spuštění ventilátoru na první rychlost. Stiskněte žluté tlačítko pro přepínání rychlostí. Stiskněte červené tlačítko pro aktivaci funkce oscilace ventilátoru. Dalším stisknutím černého tlačítka ventilátor zastavíte a vypnete.



Lekce 38 Elektronické piano

🎯 Cíle kurzu

-Pochopení fungování elektronických kláves
-Napsání programu pro tvorbu efektů pro elektronické klávesy

🔍 Úvod do kurzu

Elektronické klávesy jsou oblíbeným hudebním nástrojem. Stisknutím různých kláves vznikají zvuky různých frekvencí, které se kombinují a vytvářejí krásné melodie.

V této lekci se naučíme, jak si vyrobit „elektronickou klávesnici“, která dokáže hrát jednoduché melodie a umožní vám zažít radost z „hraní“ vlastníma rukama.

🧠 Body znalostí

1. Elektronická klávesnice

Elektronické klávesy jsou klávesový nástroj, který elektronicky produkuje zvuk. Když stisknete různé klávesy, obvod uvnitř klávesnice generuje elektrické signály o různých frekvencích. Tyto signály jsou zesíleny a poté vysílají odpovídající tóny přes reproduktory. Elektronické klávesy jsou lehké, mají bohatý tón a jsou široce používány v hudebním vzdělávání, zábavě a produkci elektronické hudby.

Jak funguje elektronická klávesnice

Základním principem elektronických kláves je převod stisknutí kláves na elektrické signály odpovídajících frekvencí a přehrávání odpovídajících tónů přes reproduktory. Základní proces je následující:

(1) Ovládání kláves

Když hráč stiskne klávesu, aktivuje se spínač uvnitř elektronické klávesnice a vyšle elektrický signál odpovídající dané klávese.

(2) Generování frekvence

Elektrický signál je zpracováván elektronickým obvodem za účelem generování elektrického signálu o specifické frekvenci. Tato frekvence odpovídá výšce zvuku, který by měl klíč vydávat.

(3) Syntéza zvuku

Syntezátor uvnitř elektronické klávesnice zpracovává tyto frekvenční signály a simuluje tak zabarvení různých nástrojů.

(4) Zesílení signálu

Zpracovaný elektrický signál je zesílen zesilovačem, aby byl zvuk dostatečně hlasitý.

(5) Zvukový výstup

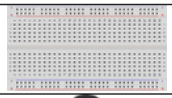
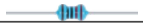
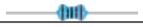
Nakonec je elektrický signál převeden reproduktorem na zvuk a hráč slyší odpovídající tón.

Experiment

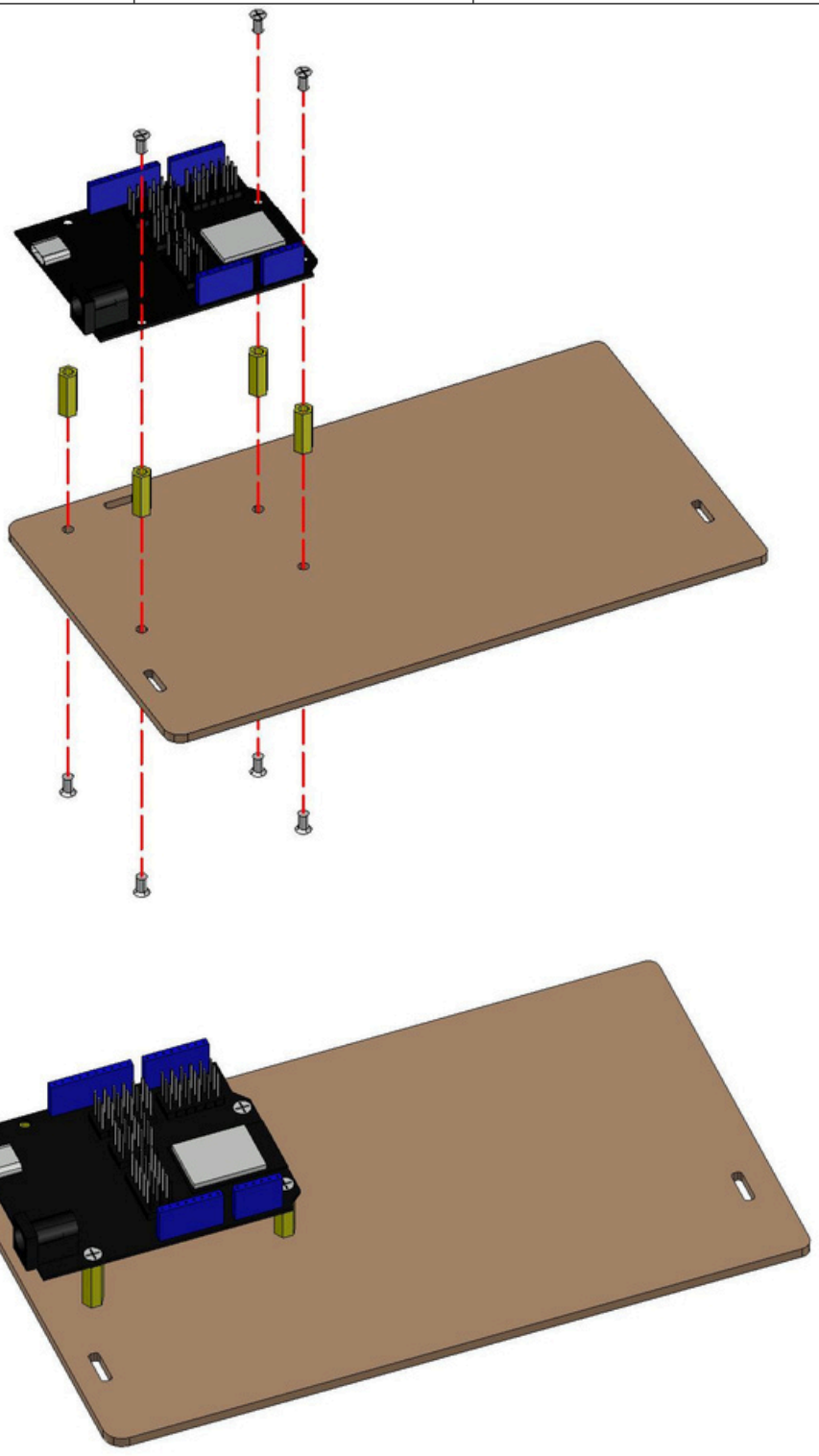
1. Popis projektu

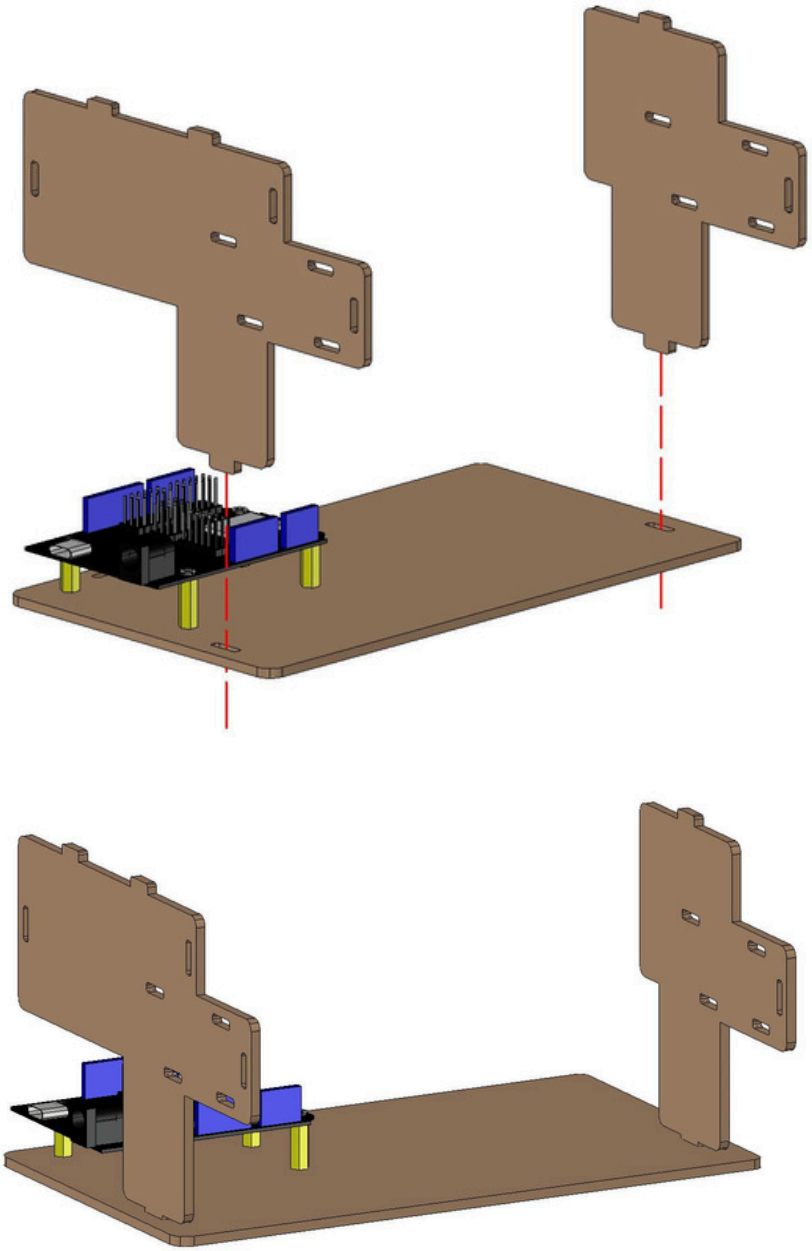
Připojením tlačítek k sedmi pinům může uživatel spustit sedm tónů (C4 až B4). Každé tlačítko odpovídá pevné frekvenci. Po stisknutí tlačítka bzučák přehraje odpovídající tón; po uvolnění tlačítka zvuk ustane.

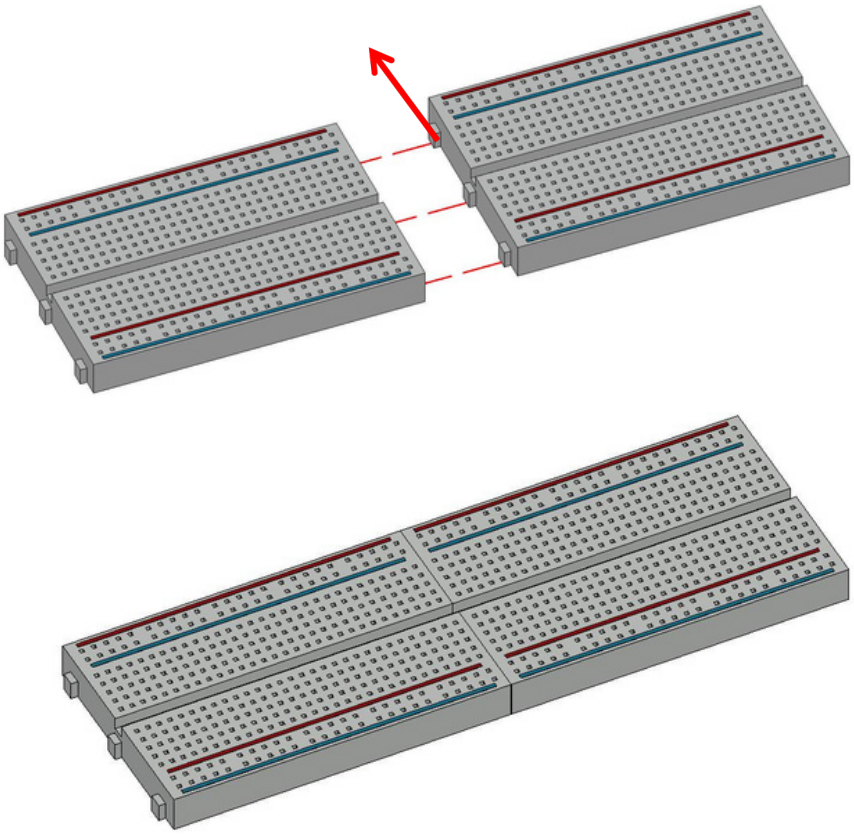
2. Seznam hardwaru

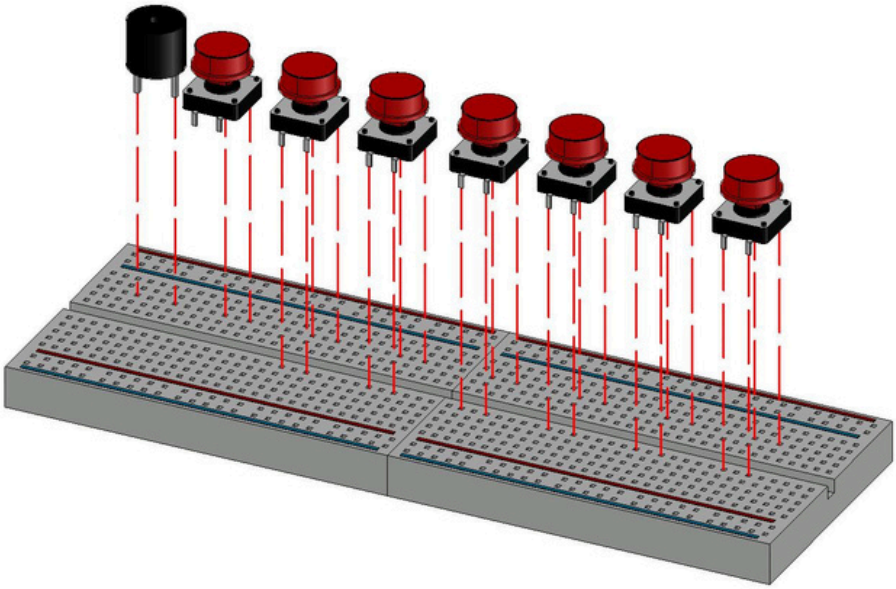
Obraz	Jméno	Množství
	Deska řadiče ESP32	1
	Nepájivé pole s 400 spojovacími body	2
	Pasivní bzučák	1
	Tlačítko	7
	Rezistor 1 kΩ	1
	Rezistor 10 kΩ	7

3. Sestavte elektronickou klávesnici (Pokud jste si nezakoupili stavební materiál, můžete krok montáže přeskočit a přejít přímo k připojení hardwaru a učení programu)

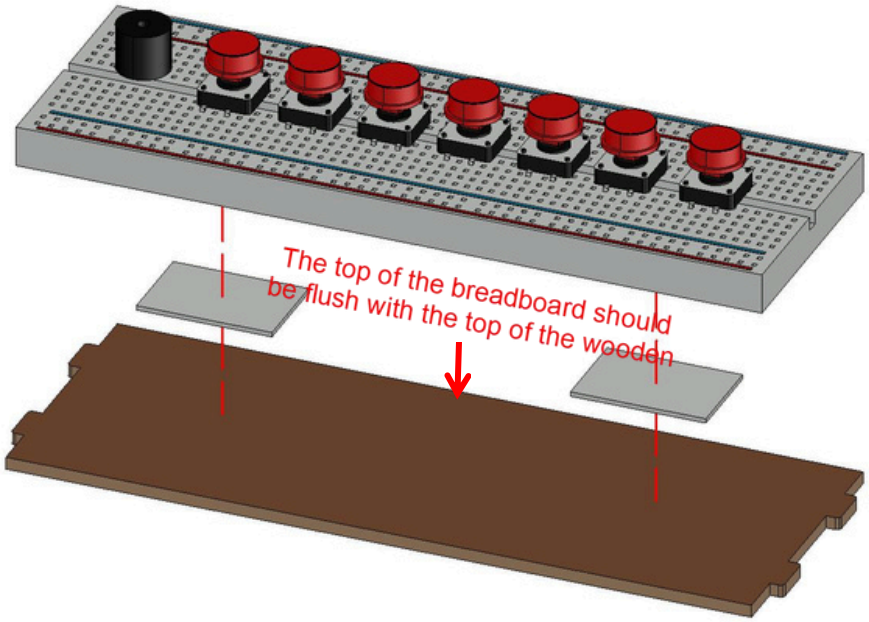
Krok 1 Nainstalujte desku řadiče ESP32			
Seznam dílů	ESP32 Ovladač Deska*1	M3*12MM Dvouprůchodový Měděný sloup*4	M3*6MM Plochá hlava Šrouby*7
Ilustrace			

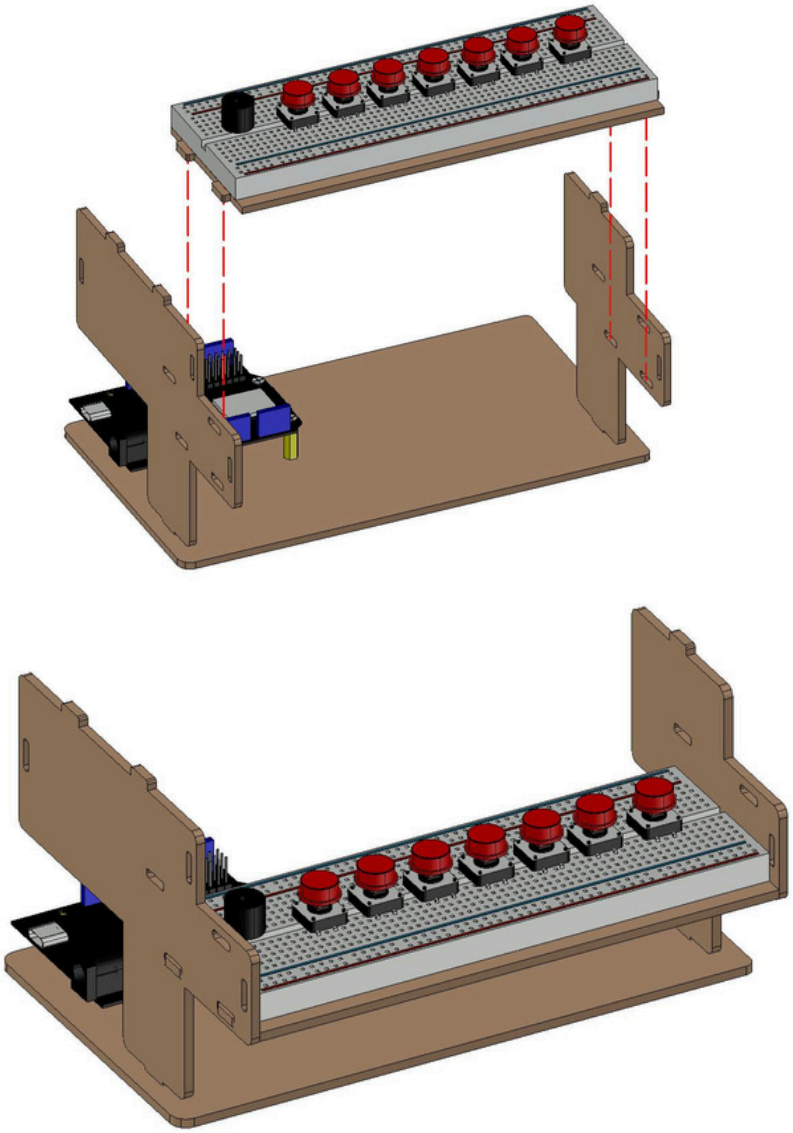
Krok 2 Instalace bočních panelů piana	
Ilustrace	

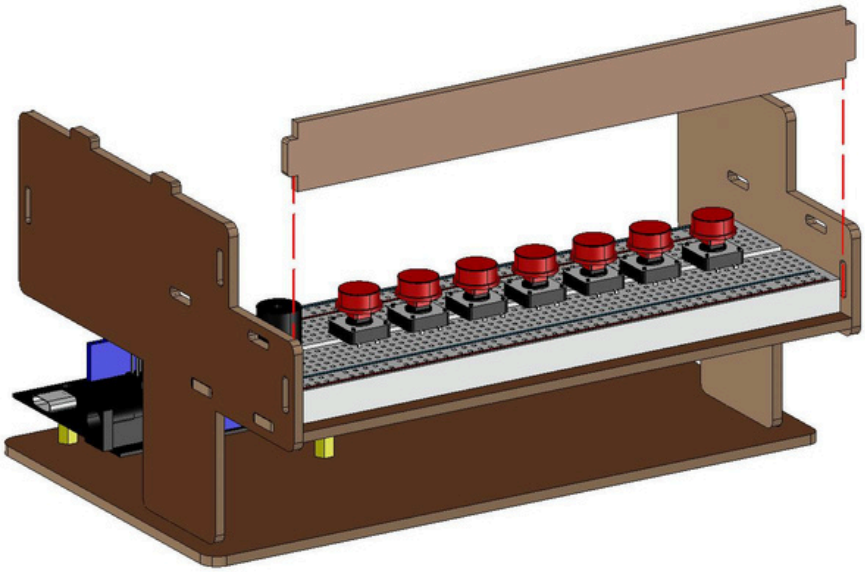
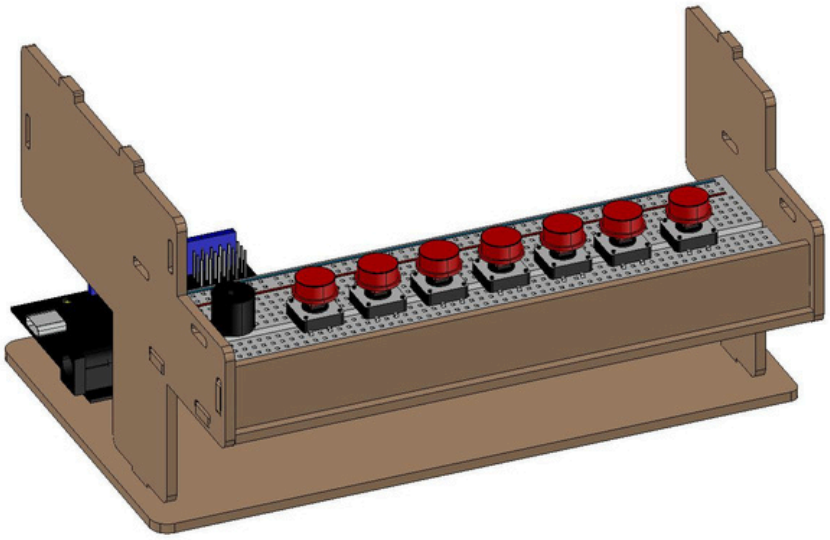
Krok 3. Sestavení nepájivého pole	
Seznam dílů	400jamkové nepájivé pole*2
Ilustrace	Poznámka: Zarovnejte obě nepájivé desky tak, aby jejich strany těsně přiléhaly k sobě. Poté jemně zacvakněte stranu s vyvýšeným výstupkem směrem nahoru do drážky druhé nepájivé desky, dokud se bezpečně nezapojí. Pro snadné zapojení udržujte napájecí kolejnice zarovnané. 

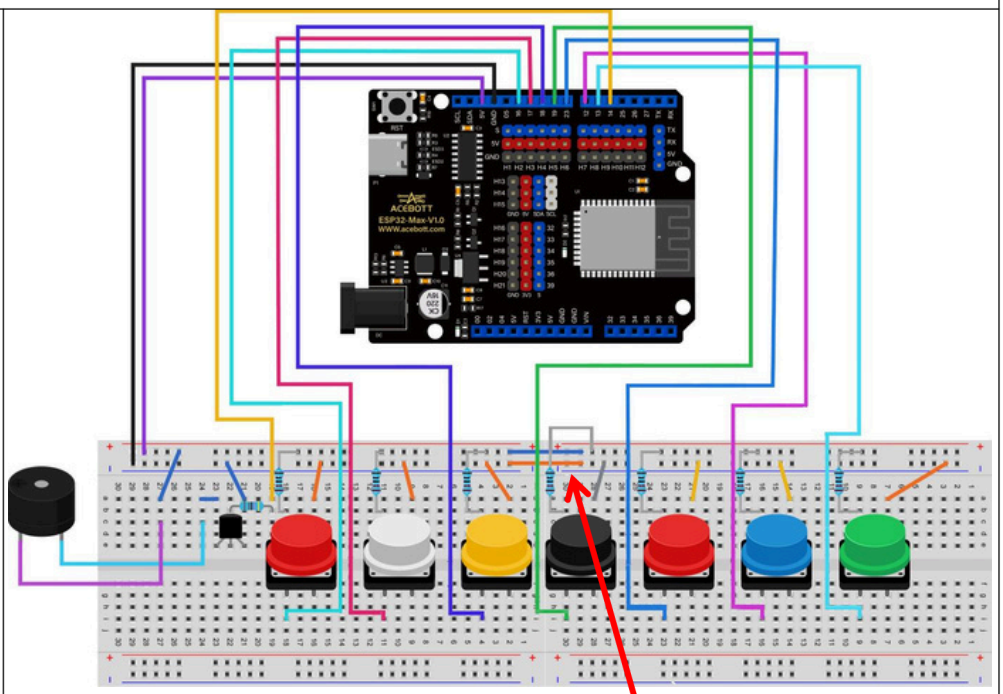
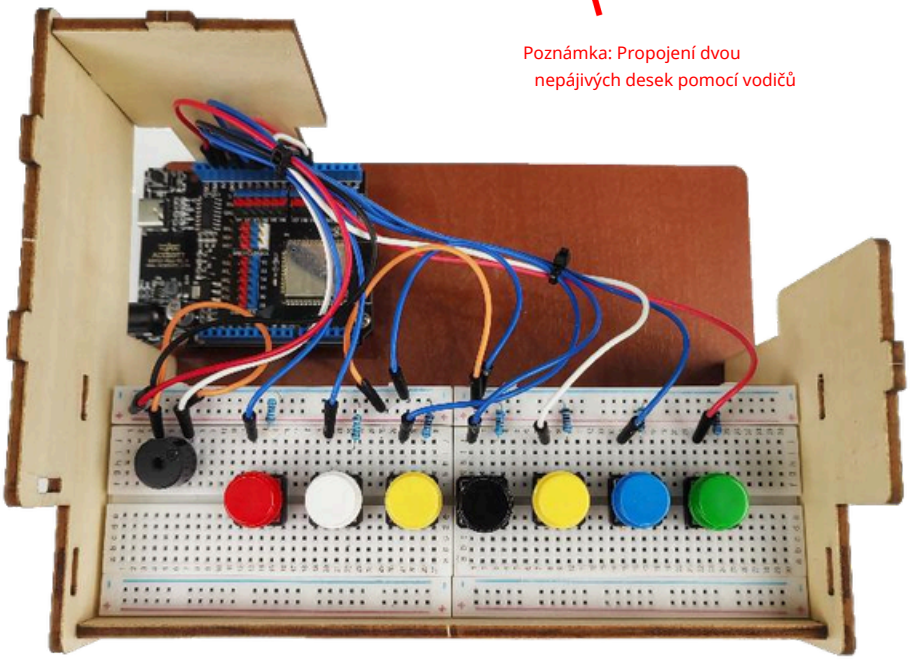
Krok 4 Instalace tlačítka a bzučáku		
Seznam dílů	Tlačítko*7	Bzučák*1
Ilustrace		

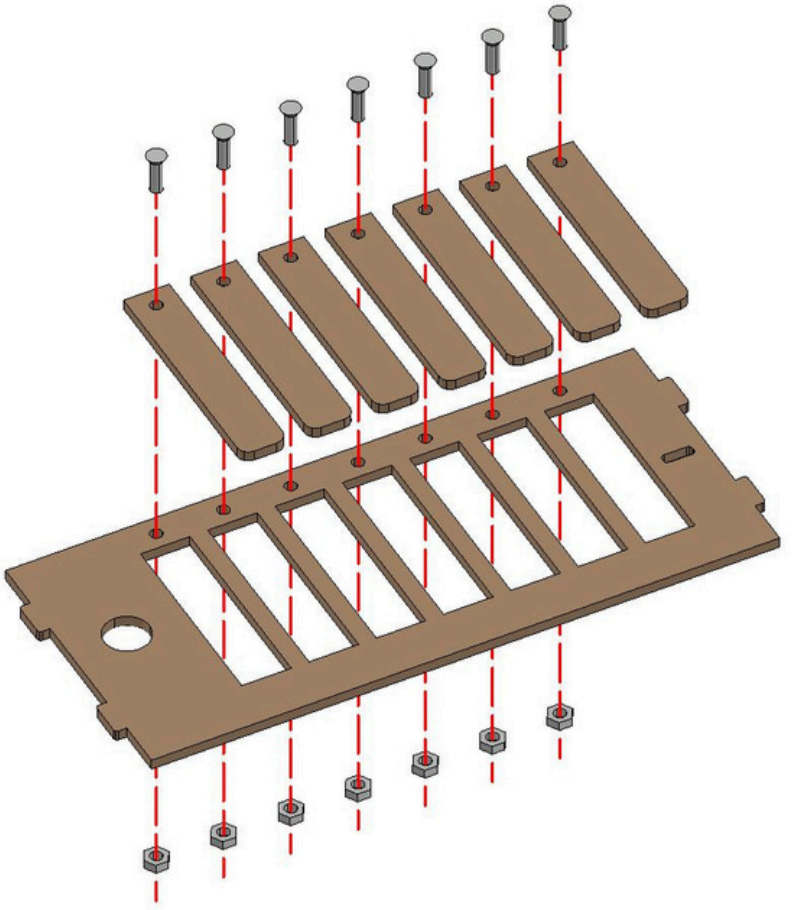
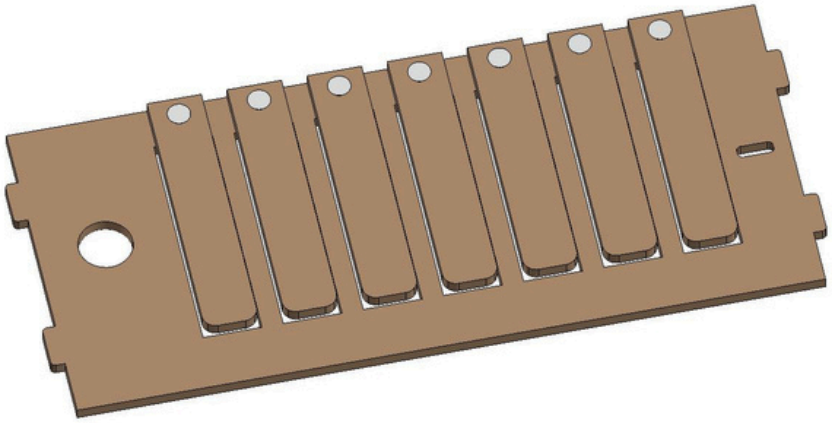
Poznámka: Tlačítka je nutné vkládat přesně na místa znázorněná na obrázku.

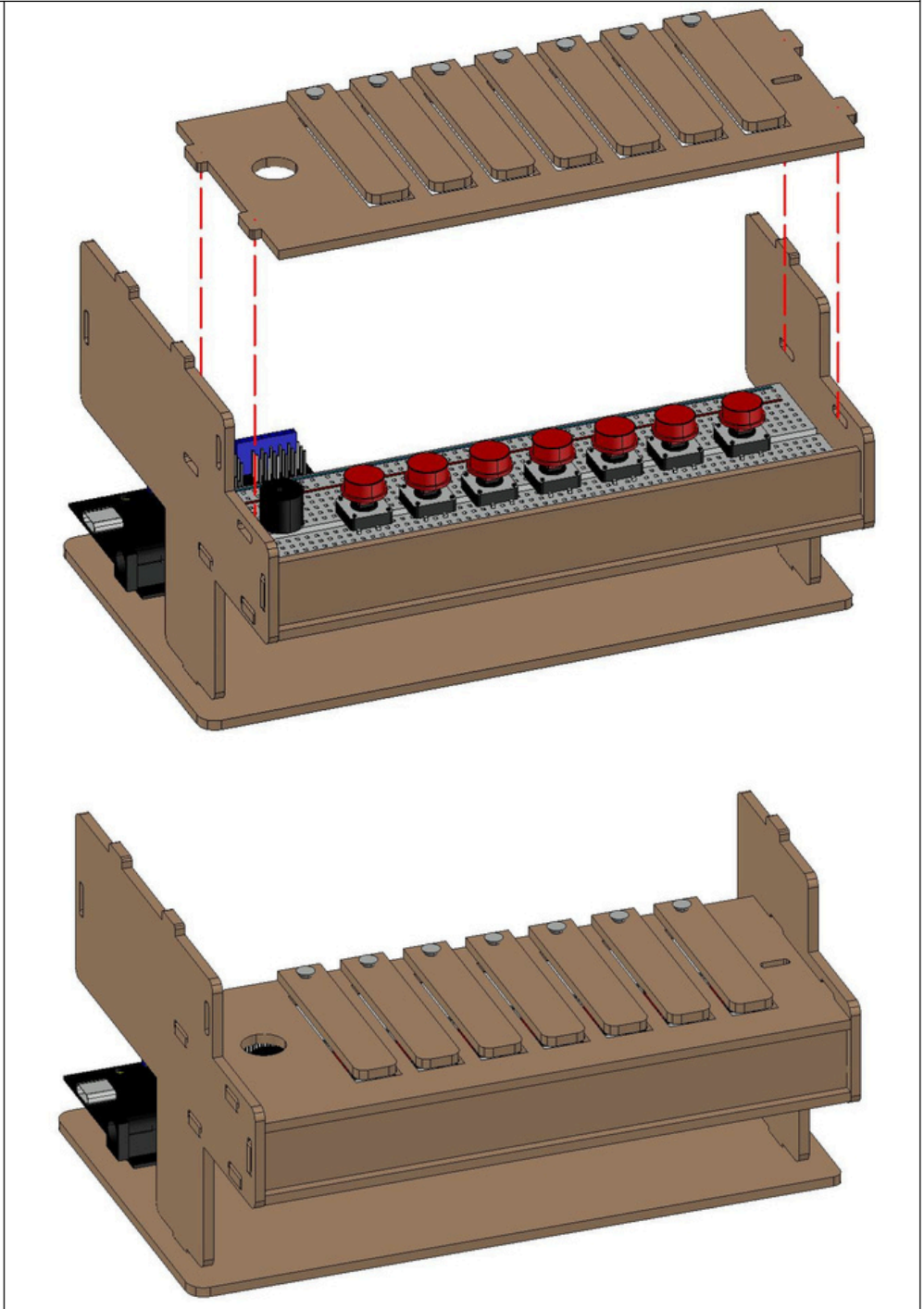
Krok 5. Zajistěte nepájivé pole	
Seznam dílů	Oboustranná páska*2
Šití Ilustrace	

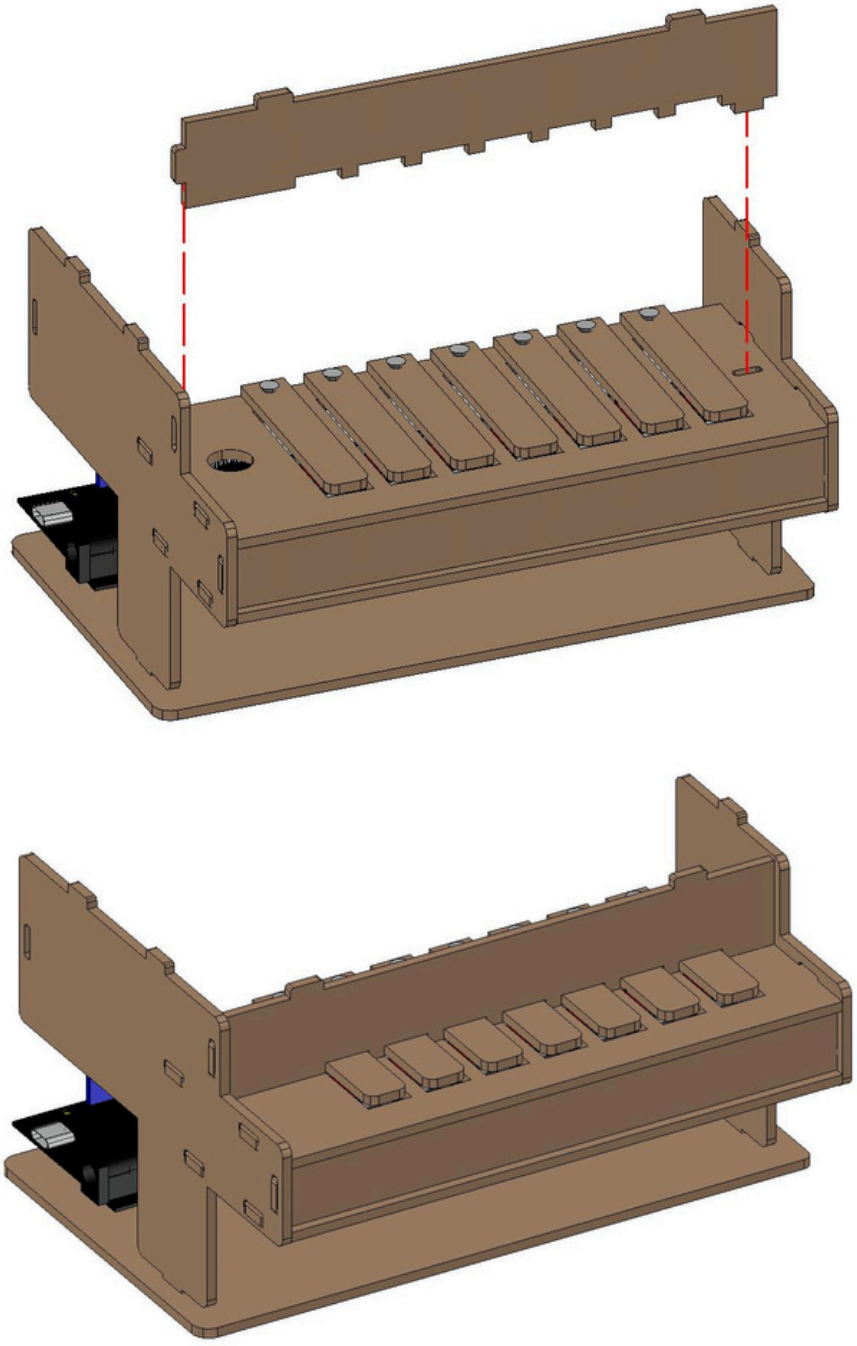
Krok 6. Instalace podpěry tlačítka	
Ilustrace	

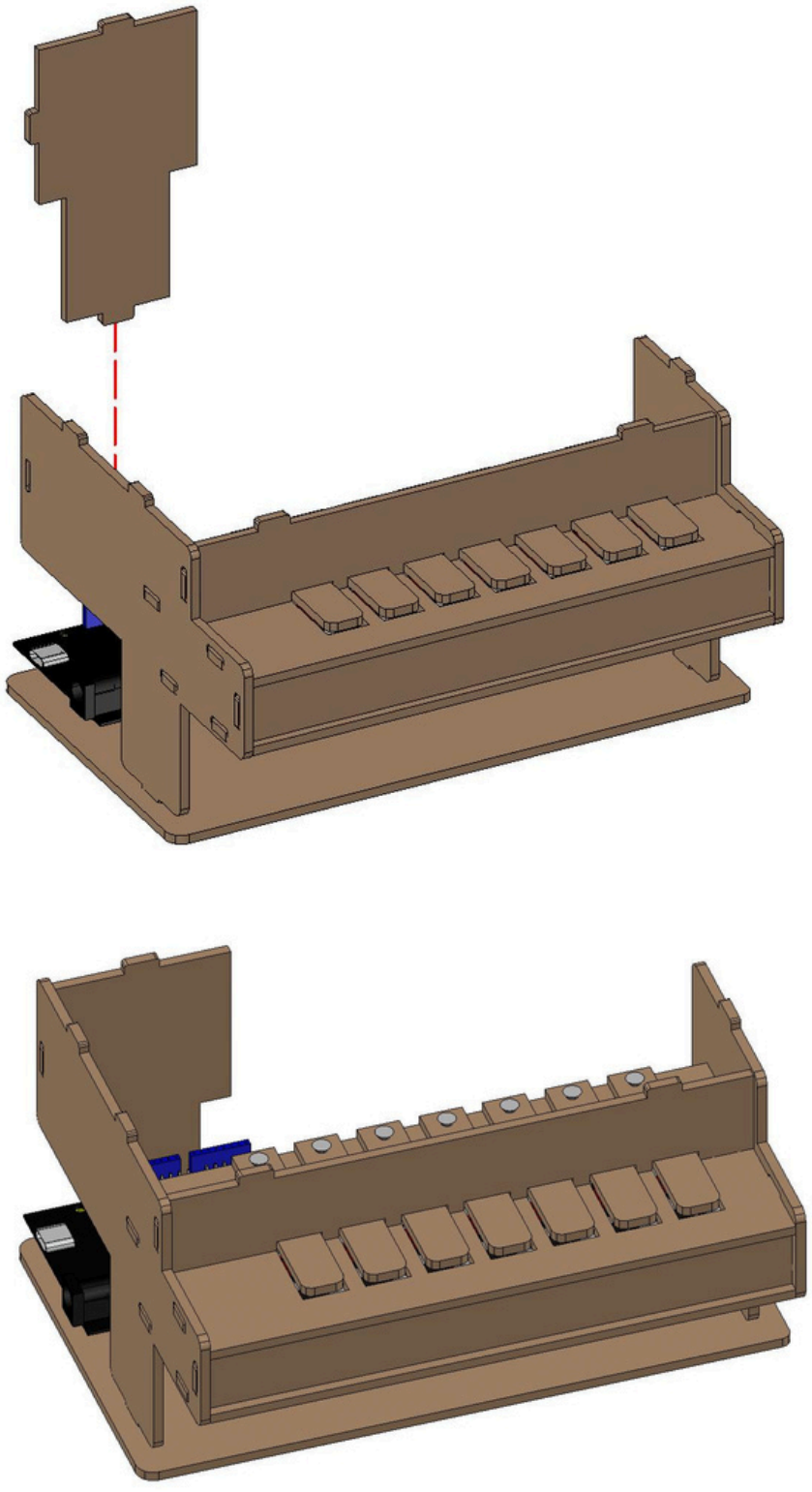
Krok 7. Instalace předního panelu piana	
Ilustrace	  **Poznámka: Než budete pokračovat, zapojte vodiče podle schématu zapojení uvedeného níže.**

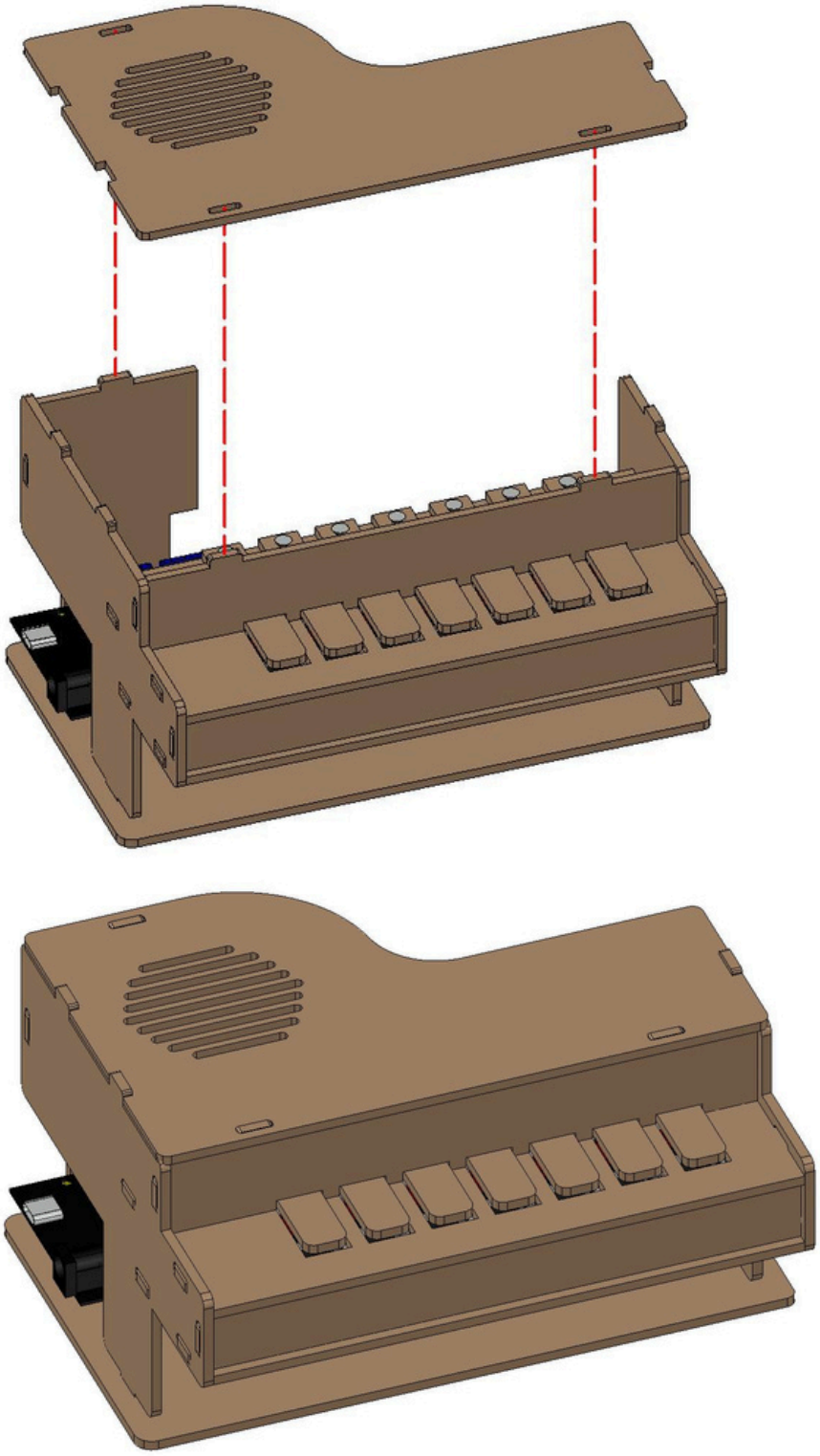
Krok 8: Zapojení	
Ilustrace	 

Krok 9 Instalace klíčů		
Seznam dílů	M3*6mm plochá hlava Šroub*7	Matice M3*7
Ilustrace	 	

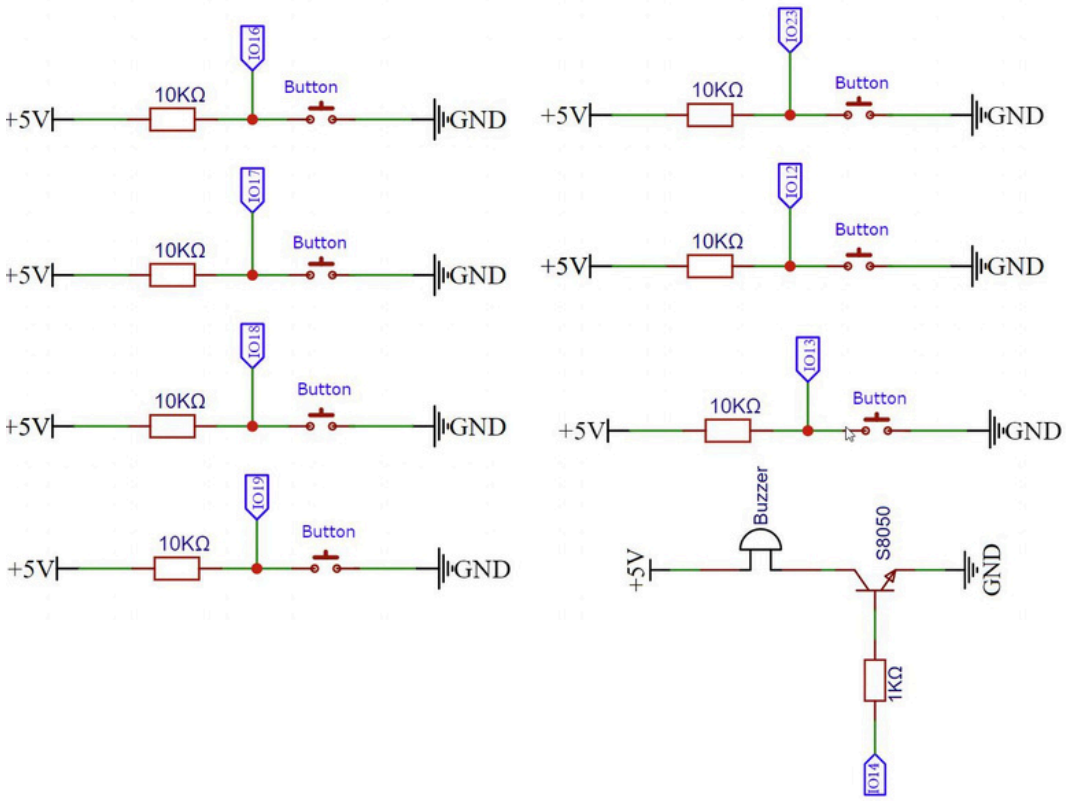
Krok 10 Instalace klávesnice	
Ilustrace	

Krok 11 Instalace zvedáku klávesnice	
Ilustrace	

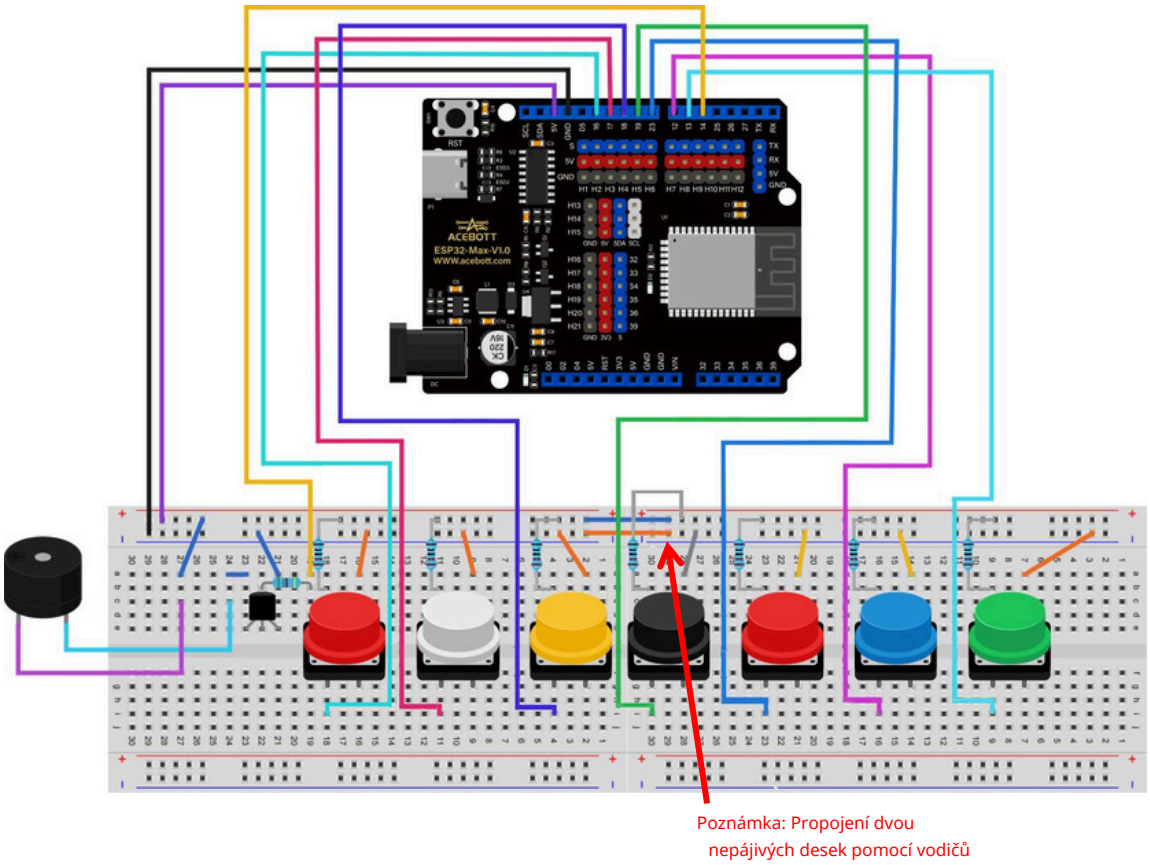
Krok 12 Nainstalujte zadní panel	
Ilustrace	

Krok 13 Nainstalujte kryt klávesnice	
Ilustrace	

4. Schéma zapojení



5. Hardwarová připojení



Poznámka: Propojení dvou nepájivých desek pomocí vodičů

6. Referenční postupy

Otevřete "Electronic_Piano.ino " soubor v adresáři "English\Arduino(Experienced Learner)\2.Code\lesson 38 Electronic keyboard\Electronic_Keyboard" nebo zkopírujte následující kód do Arduino IDE. Po výběru vývojové desky (ESP32 Dev Module) a příslušného portu podle dříve naučených kroků klikněte  nahrát program.

```
#define BUZZER_PIN 14

int keys[] = {16, 17, 18, 19, 23, 12, 13}; // Key input pins
int notes[] = {523, 587, 659, 698, 784, 880, 988}; // Frequencies for C4,
D4, E4, F4, G4, A4, B4

void setup() {
for (int i = 0; i < 7; i++) {
pinMode(keys[i], INPUT_PULLUP); // Set each key pin as input with internal
pull-up resistor
}
}

void loop() {
for (int i = 0; i < 7; i++) {
if (digitalRead(keys[i]) == LOW) { // Check if key is pressed (active low)
tone(BUZZER_PIN, notes[i]); // Play the corresponding tone frequency on
buzzer pin
while(digitalRead(keys[i]) == LOW); // Wait here until key is released
noTone(BUZZER_PIN); // Stop playing the tone
}
}
}
```

Po úspěšném nahrání programu stiskněte klávesy pro přehrání odpovídajících not a bzučák vydá odpovídající zvuk.