

Řada vzdělávacích řešení

Vzdělávací sada pro inteligentní dopravu úroveň 1



— Programování v Arduino —

Obsah

Důležité informace před použitím	4
Stavba kostry chytré továrny	8
Osvětlení ulice.....	8
Inteligentní pouliční osvětlení (I).....	14
Inteligentní pouliční osvětlení (II)	18
Základní semafore	19
Semafore s odpočítáváním	22
Inteligentní semafore	26
Inteligentní bezpečnostní ochrana pro přechod	30
Dopravní světla na zemi	38
Bezpečnostní připomenutí přechodu	41
Tlačítkový semafor	44
Inteligentní přístupová kontrola.....	48
Zobrazení parkovacích míst	58
Hlasový systém vysílání	61
Inteligentní parkovací systém integrován.....	66
Kapitola 1 Inteligentní doprava.....	72
Sekce 1: přehled inteligentní dopravy	73
Sekce 2: Mozek inteligentní dopravy	81
Jak nainstalovat Arduino IDE na Windows	85

Jak nainstalovat Arduino IDE na MacOS.....	95
Kapitola 2 Chytré pouliční osvětlení	115
Sekce 1: Pouliční osvětlení	116
Sekce 2: Inteligentní pouliční osvětlení (II)	128
Sekce 3: Inteligentní pouliční osvětlení (III)	138
Kapitola 3 Inteligentní semaforey	147
Sekce 1: Základní semaforey	148
Sekce 2: Odpočítávání semaforů	157
Sekce 3: Inteligentní semaforey	167
Kapitola 4 Chytré přechody	177
Sekce 1: Inteligentní ochrana přechodů	178
Sekce 2: Pozemní dopravní světla	186
Sekce 3: Upozornění na bezpečnost přechodu	196
Sekce 4: Tlačítková dopravní světla	205
Kapitola 5 Chytré parkoviště	215
Sekce 1: Inteligentní kontrola přístupu.....	216
Sekce 2: Zobrazení parkovacího místa	226
Sekce 3: Hlasový vysílací systém.....	236
Sekce 4: Integrovaný systém chytrého parkoviště	247
FAQ.....	258
Sleduj nás.....	263

Důležité informace před použitím

1. Vhodná baterie

Vzhledem k logistickým omezením nemůžeme dodávat potřebné baterie spolu s výrobkem. Omlouváme se za způsobené nepříjemnosti. Abychom vám pomohli rychle najít vhodnou baterii, sestavili jsme několik doporučených odkazů na Amazonu, které vám mohou pomoci při výběru.

(1) Odkaz na baterii 18650: [Klikněte zde pro získání odkazu.](#)

Poznámka: Tento produkt vyžaduje dodatečné zakoupení dvou 18650 baterií s ostrým hrotem; jeden nabíječka (nabíječka se zakupuje v závislosti na verzi ovládací desky ESP32).

(2) Návod k výběru nabíječky: Ne siž vyberete nabíječku pro baterie, ověřte si typ své řídicí desky ESP32. Pokud má zadní strana desky nápis “**Max V3.0**”, znamená to, že deska má integrovanou funkci nabíjení baterií. Můžete tedy baterie nabíjet přímo pomocí řídicí desky a **není třeba kupovat samostatnou nabíječku**. Pokud na zadní straně desky najdete označení “**Max V1.0**”, znamená to, že deska nemá funkci nabíjení baterií. V takovém případě budete potřebovat **koupit samostatnou nabíječku** pro nabíjení baterií, kterou můžete koupit pomocí odkazu na nabíječku níže.



(3) Odkaz na nabíječku: <https://shop.acebott.com/products/acebott-us-plug-eu-plug-uk-plug-dual-charger-for-18650-14500-16430-rechargeable-li-ion-battery?variant=42992073015433>

(4) Návod na nabíjení baterií pomocí řídicí desky ESP32 MAX V3.0:

Připojte bateriovou schránku k rozhraní DC napájení na řídicí desce, vypněte bateriovou schránku a pomocí datového kabelu připojte řídicí desku k počítači, aby začalo být baterie nabíjena. Počet prvních čtyř modrých indikátorů na řídicí desce ukazuje aktuální hladinu nabitosti baterie. Poslední červený indikátor ukazuje stav nabíjení: pokud svítí, znamená to, že baterie je v režimu nabíjení; pokud je vypnutý, znamená to, že baterie je úplně nabitá nebo není v režimu nabíjení.

Poznámka: 1. Tato řídicí deska podporuje pouze napájení lithiové baterie, nepoužívejte suché baterie. 2. Před nabíjením nahrajte

prázdný program do ovládací desky ESP32 a [kliknutím sem získáte prázdný program.](#)

2.Podpora po prodeji

V případě jakýchkoliv problémů kontaktujte náš tým podpory prostřednictvím e-mailu na support@acebott.com, odpovíme vám do 24 hodin. Můžete také naskenovat níže uvedený QR kód a sledovat nás pro získání návodů na řešení problémů a nejnovějších informací.



ACEBOTT FB QR Code YouTube QR Code

3. Video instalace softwaru

Nainstalujte Arduino IDE: <https://youtu.be/VwjsLL4zJd4>

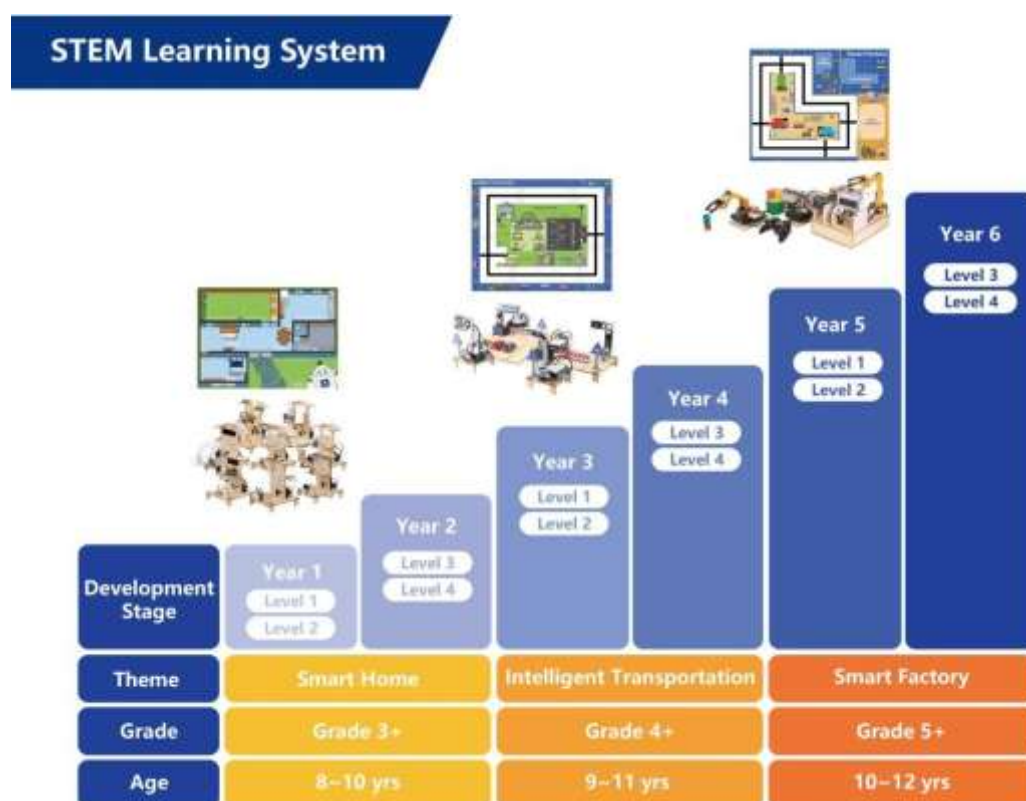
Nainstalujte ovladač CH340: <https://youtu.be/B0afbif4jXA>

Nainstalujte esp32: <https://youtu.be/VcfxWm-rZLs>

Nainstalujte knihovny: <https://youtu.be/BIKFr2oVEVQ>

4. Další vzdělávací série

Tento produkt je součástí vzdělávací série ACEBOTT. Tato série zahrnuje tři základní tematické oblasti: chytrý dům, chytrá doprava a chytrá továrna. Každá oblast obsahuje několik sad, které jsou vzájemně propojeny. Pokud chcete získat komplexní zkušenosti s programováním, sledujte podrobnosti o dalších produktech v rámci této vzdělávací série.



Educational Products



QE023

Smart Home Education Kit- Level 1

- ① Intelligent Lighting
- ② Intelligent Entertainment
- ③ Intelligent Temperature & Humidity Control



QE024

Smart Home Education Kit- Level 2

- ① Automatic Pet Feeder
- ② Automatic Drying Rack
- ③ Intelligent Safety Alarm System
- ④ Intelligent Access Control
- ⑤ Automatic Watering System



QE035

Smart Home Education Kit- Year 1

- ① Intelligent Lighting
- ② Intelligent Entertainment
- ③ Intelligent Temperature & Humidity Control
- ④ Automatic Pet Feeder
- ⑤ Automatic Drying Rack
- ⑥ Intelligent Safety Alarm System
- ⑦ Intelligent Access Control
- ⑧ Automatic Watering System



QE027

Intelligence Transportation Education Kit - Level 1

- ① Smart Street Light
- ② Smart Traffic Light
- ③ Smart Crossing
- ④ Smart Parking Lot



QE031

Smart Factory Education Kit - Level 1

- ① Smart Robot Arm
- ② Smart Conveyer
- ③ Smart Warehouse System

More educational products are coming online, stay tuned!

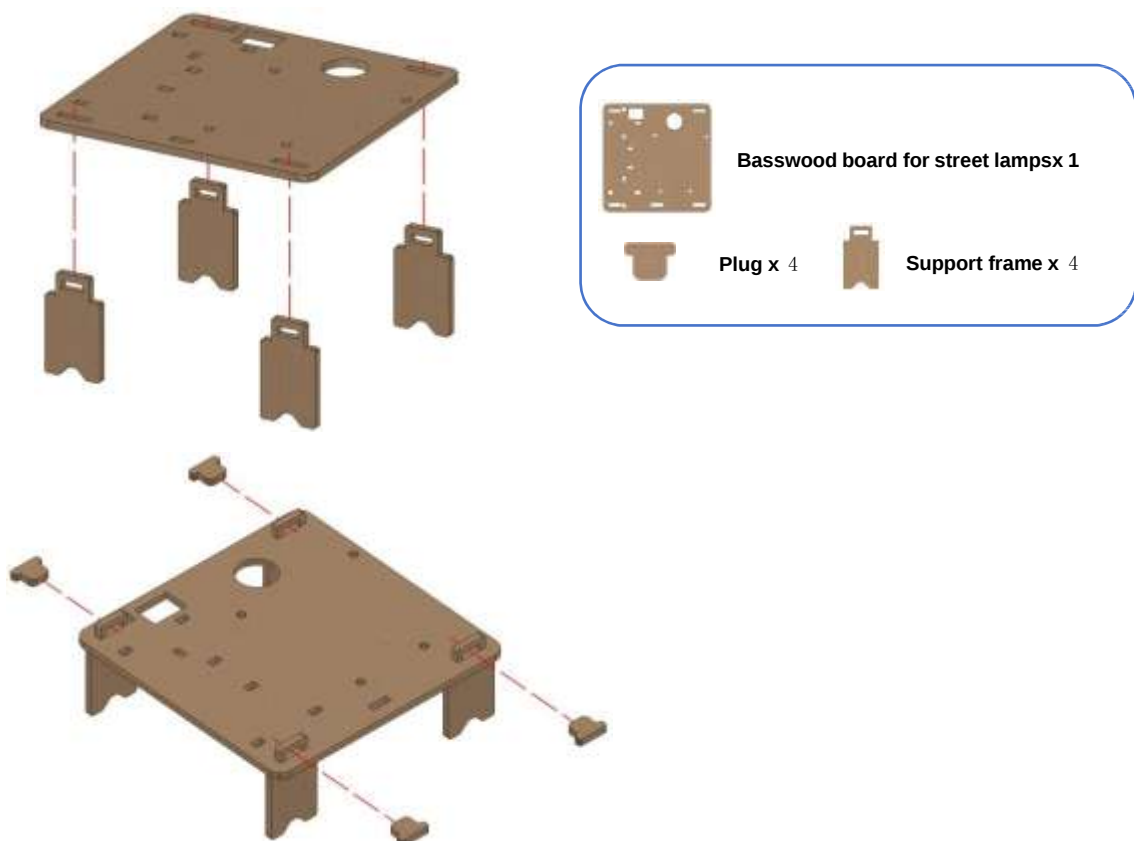
Stavba kostry chytré továrny

Upozornění:

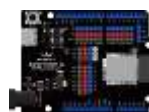
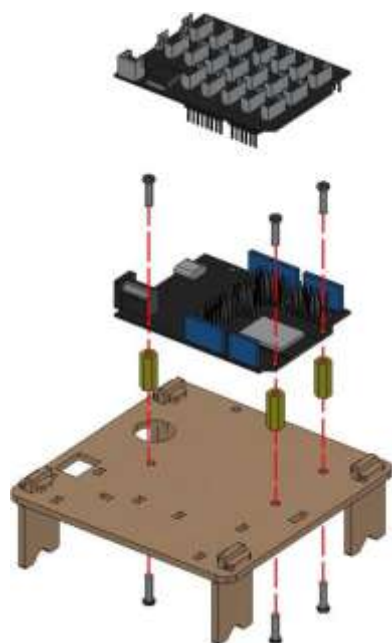
1. Tento návod vyžaduje 2 baterie 18650 pro napájení, které si musíte zajistit sami.
2. Pokud odkaz na program v dokumentu nelze otevřít, můžete ve stažených výukových materiálech najít cestu "Čeština\Arduino(Experienced Learner)\3.Arduino program" a otevřít příslušný program.

01 Ulicí osvětlení

Krok 1: Sestavte základnu.



Krok 2: Nainstalujte řídicí desku ESP32 a expanzní desku.



ESP32 x 1



Expansion board x 1



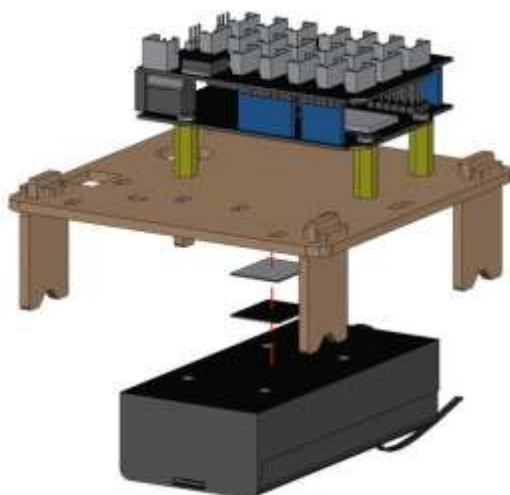
M3*6 MM Flat Head Screw x 6



Copper Column x 3

Poznámka: Nejprve upevněte měděný sloupek na základnu, poté nainstalujte řídicí desku ESP32 na měděný sloupek a nakonec nainstalujte expanzní desku. Při instalaci expanzní desky se ujistěte, že díry jsou správně zarovnány s ESP32 před jejím upevněním.

Krok 3: Instalace držáku baterie.



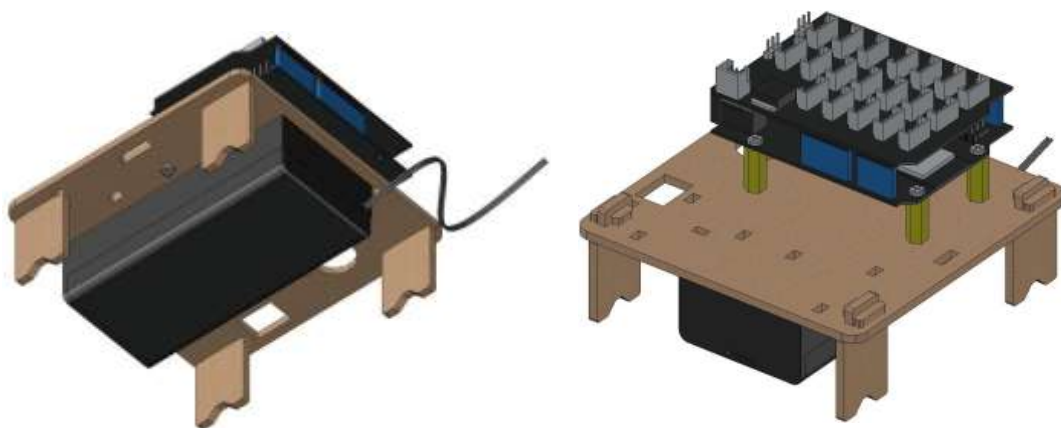
18650 Battery Holder x 1



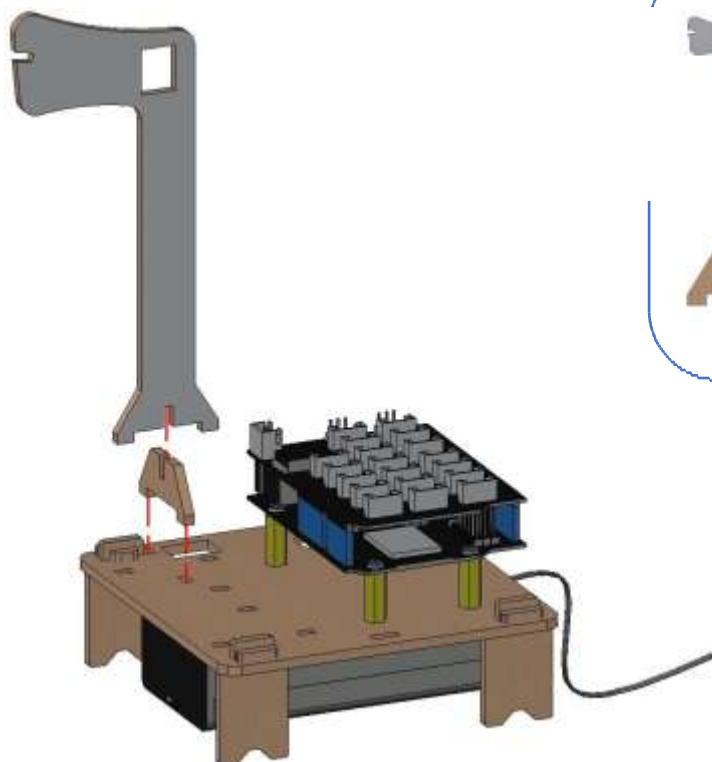
20mm Double Sides Tape x 1

Poznámka: Použijte oboustrannou lepicí pásku k připevnění držáku baterie na spodní stranu základny a nezakrývejte elektrické vodiče a

otvory pro piny na základně.



Krok 4: Instalace držáku pouliční lampy.



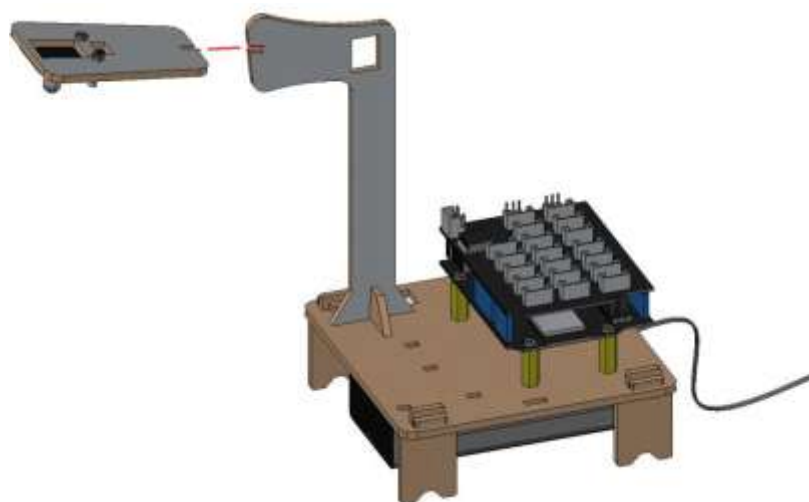
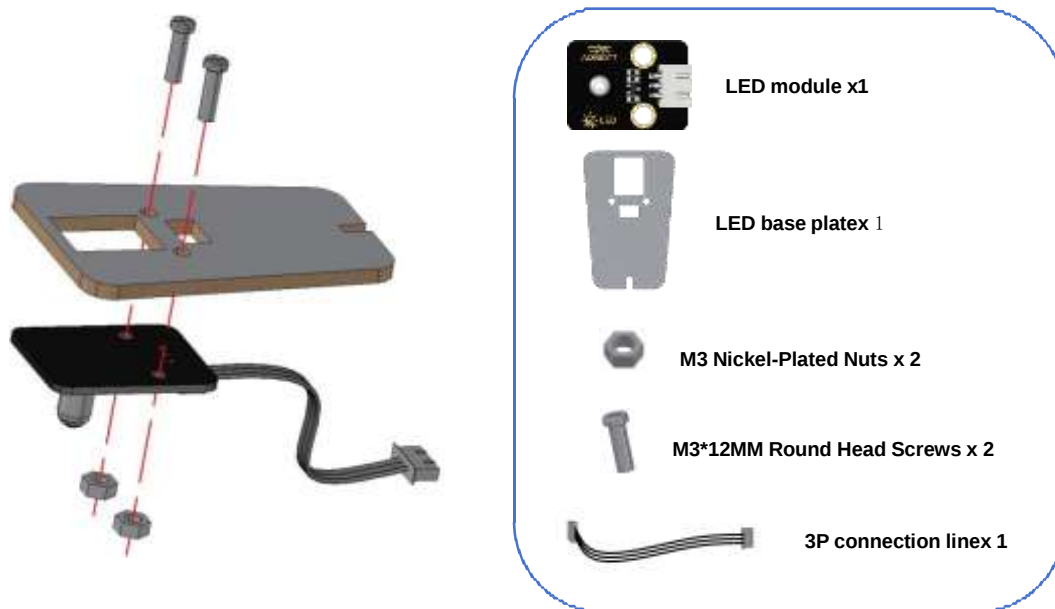
Basswood board for street lamp bracketsx 1

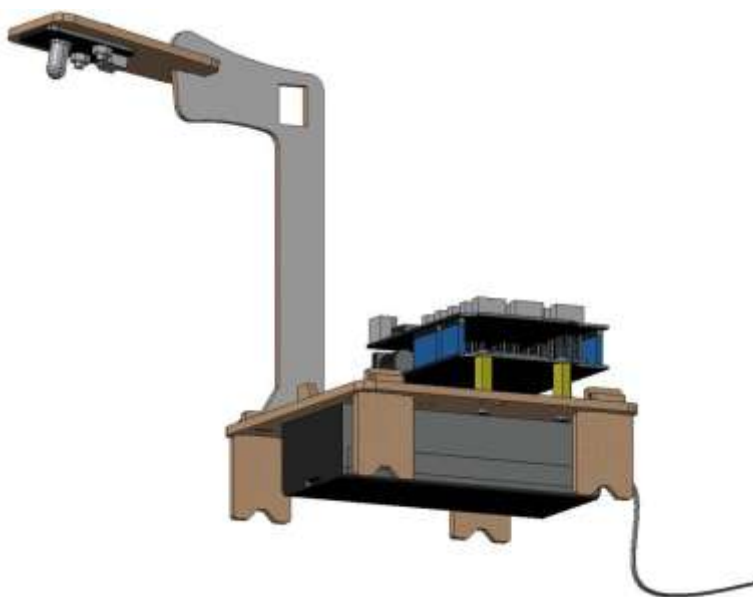


Bracket boardx 1

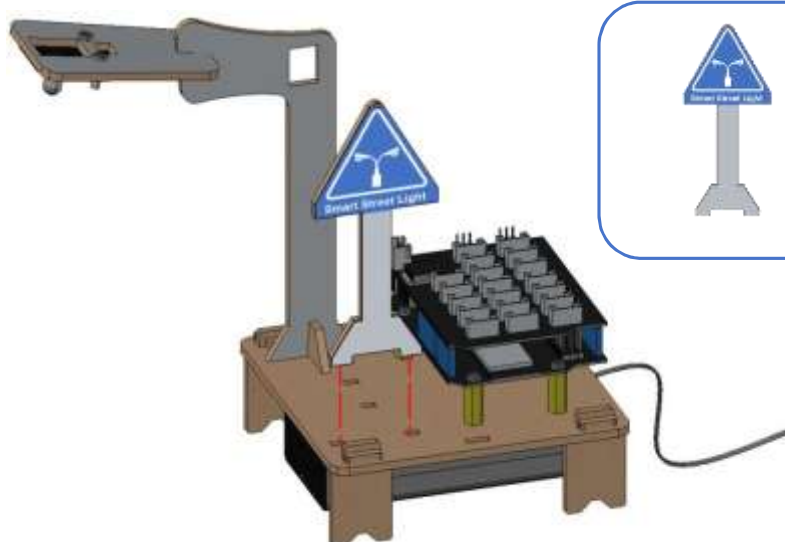
Krok 5: Instalace LED modulů.

Poznámka: Před instalací LED modulu se ujistěte, že jste předem připojili 3-pinový senzorový kabel.

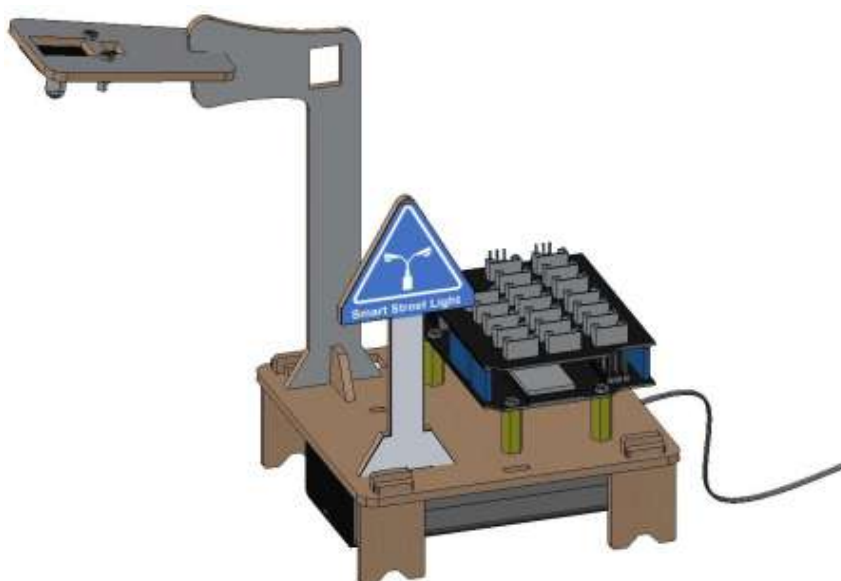




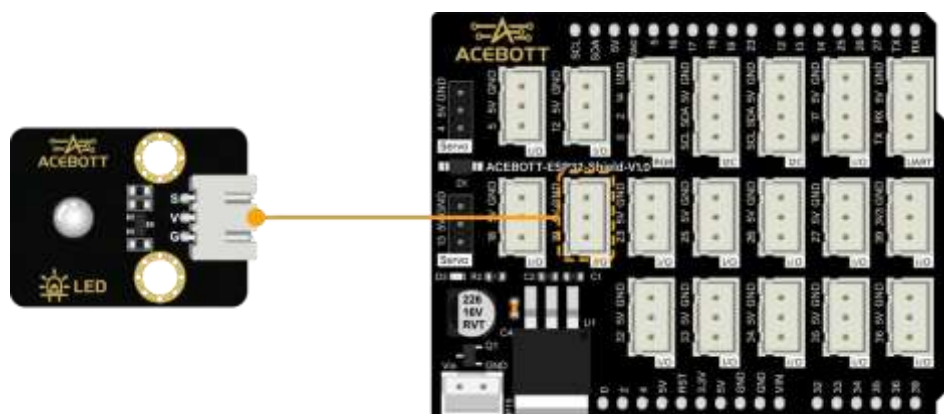
Krok 6: Instalace stojící karty na základnu.



**Intelligent street
light signage x 1**



Krok 7: Zapojení.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
LED module	S	Pin19
	V	5V
	G	GND

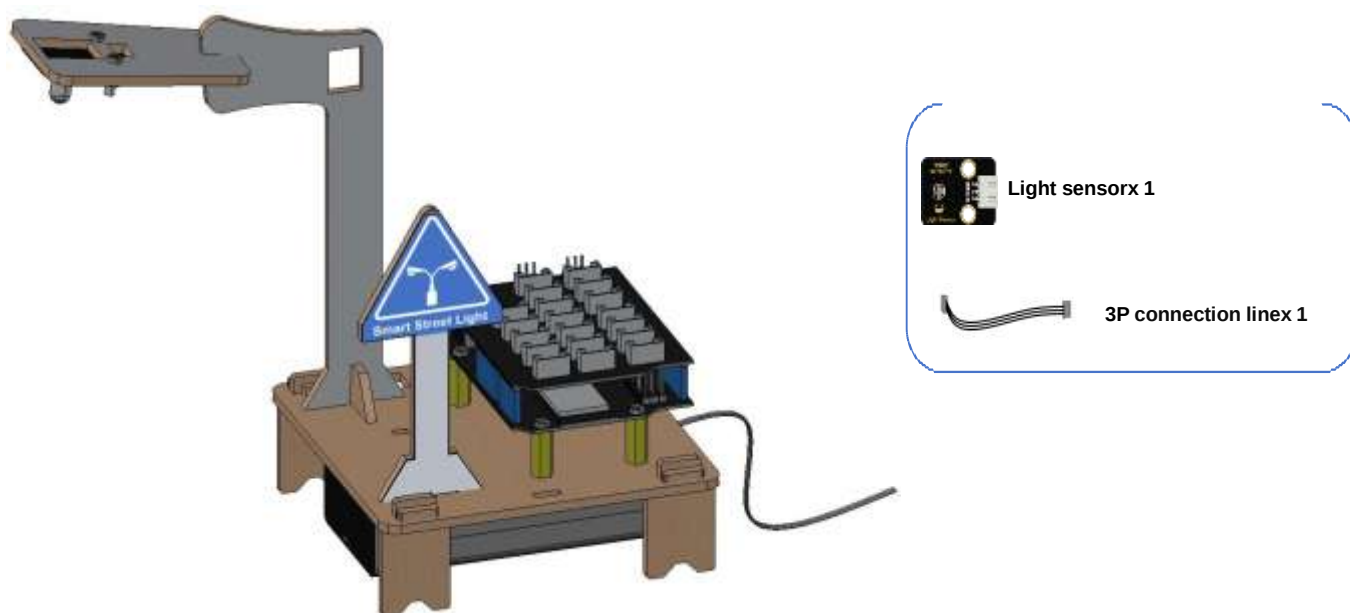
Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

02 Inteligentní pouliční osvětlení (I)

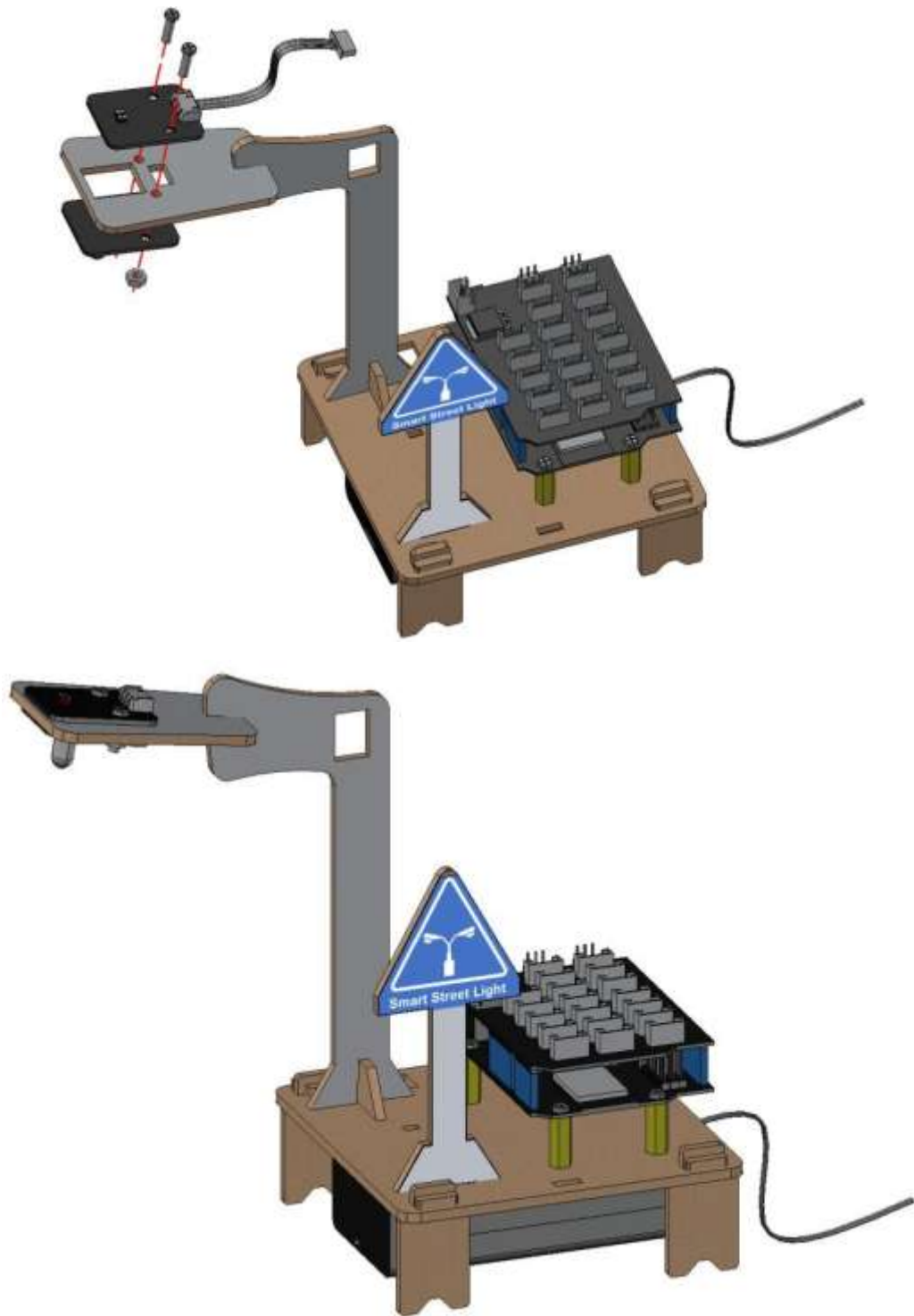
Upozornění: Tento projekt bude následovat strukturu

„pouličního osvětlení“ (jak je uvedeno na obrázku níže). Konkrétní konstrukční kroky zde nebudou uvedeny. Začneme přímo instalací světelného senzoru.



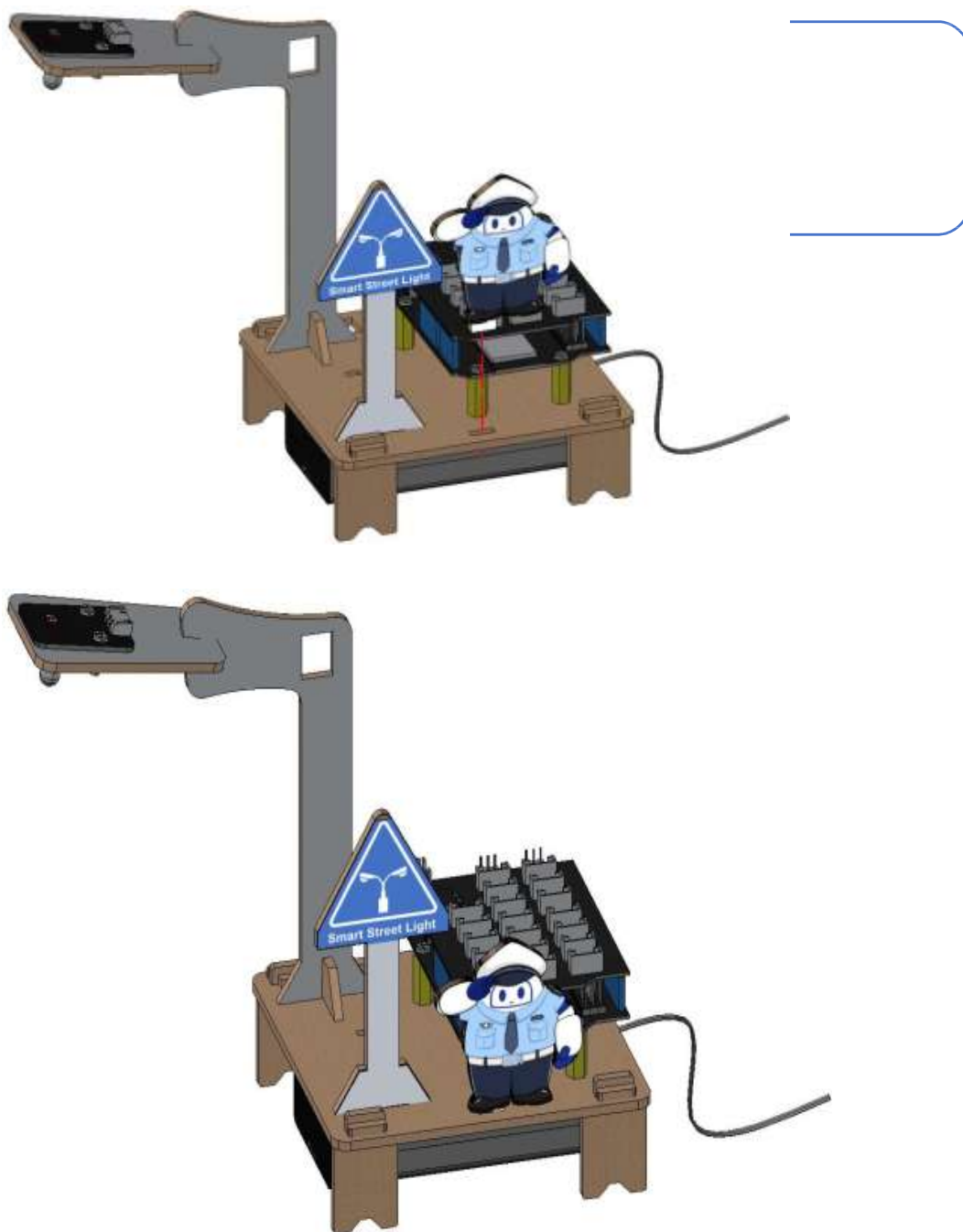
Krok 1: Instalace světelného senzoru.

Před instalací světelného senzoru předem připojte 3pinový kabel senzoru a odstraňte šrouby z LED modulu. Poté použijte odstraněné šrouby k upevnění LED modulu a světelného senzoru dohromady.

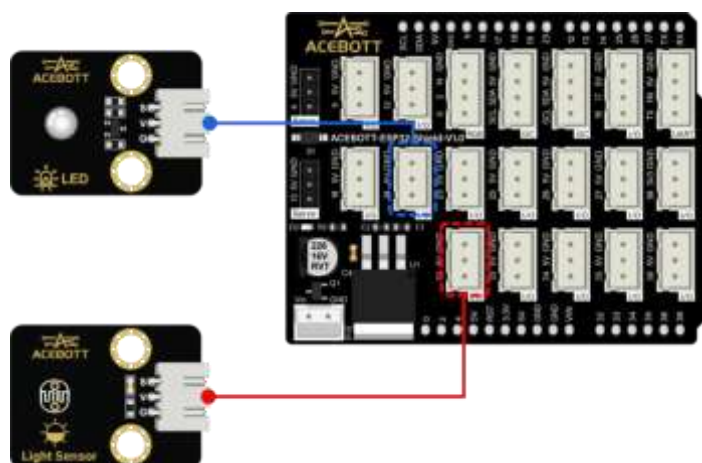


Krok 2: Instalace LUMI.

LUMI standing cardx 1



Krok 4: Zapojení.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
LED module	S	Pin19
	V	5V
	G	GND
Light sensor	S	Pin32
	V	5V
	G	GND

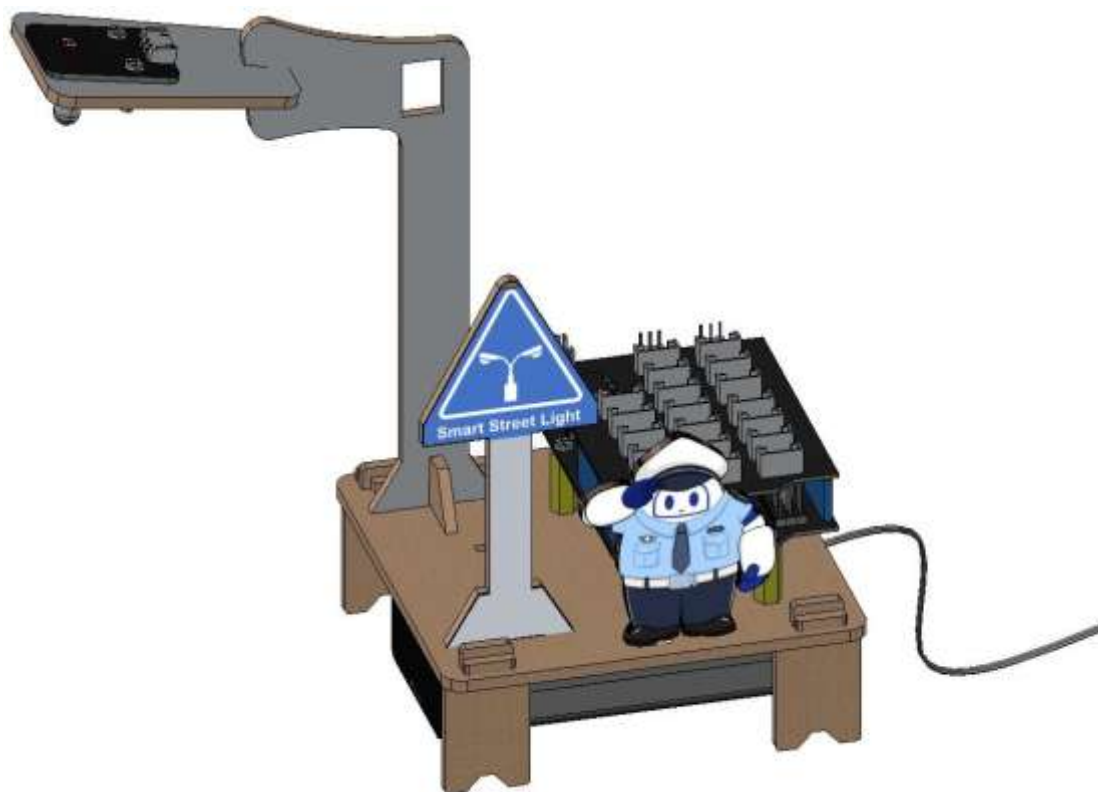
Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně

dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

03 Inteligentní pouliční osvětlení (II)

Upozornění: Struktura Inteligentního pouličního osvětlení (II) je stejná jako u Inteligentního pouličního osvětlení (I) (viz obrázek níže). Postupujte podle kroků uvedených pro Inteligentní pouliční osvětlení (I).



[Klikněte zde pro otevření programu pro tento projekt.](#)

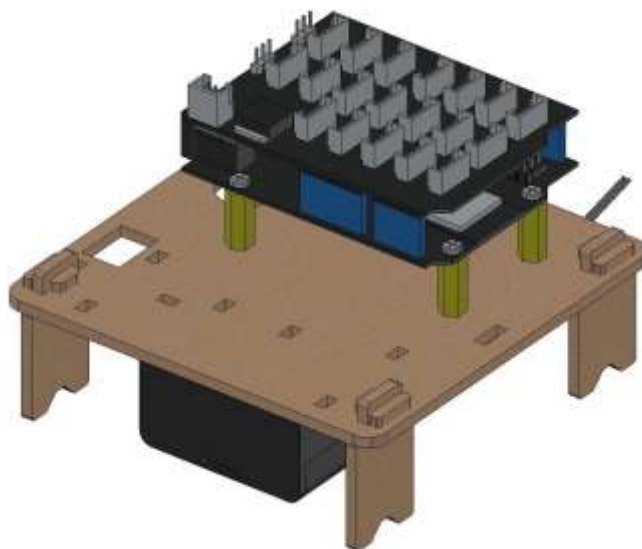
04 Základní semaforey

Upozornění: Tento projekt bude využívat základní strukturu

„Inteligentního pouličního osvětlení“ (viz obrázek níže).

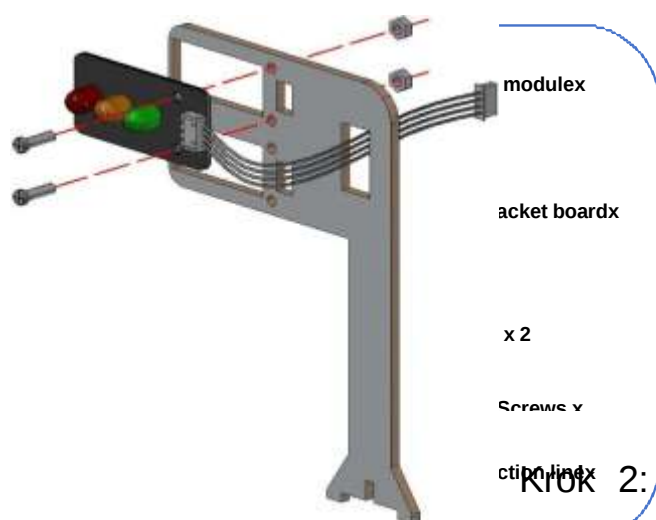
Konkrétní kroky konstrukce zde nejsou uvedeny.

Přejdeme rovnou k instalaci modulu semaforu.

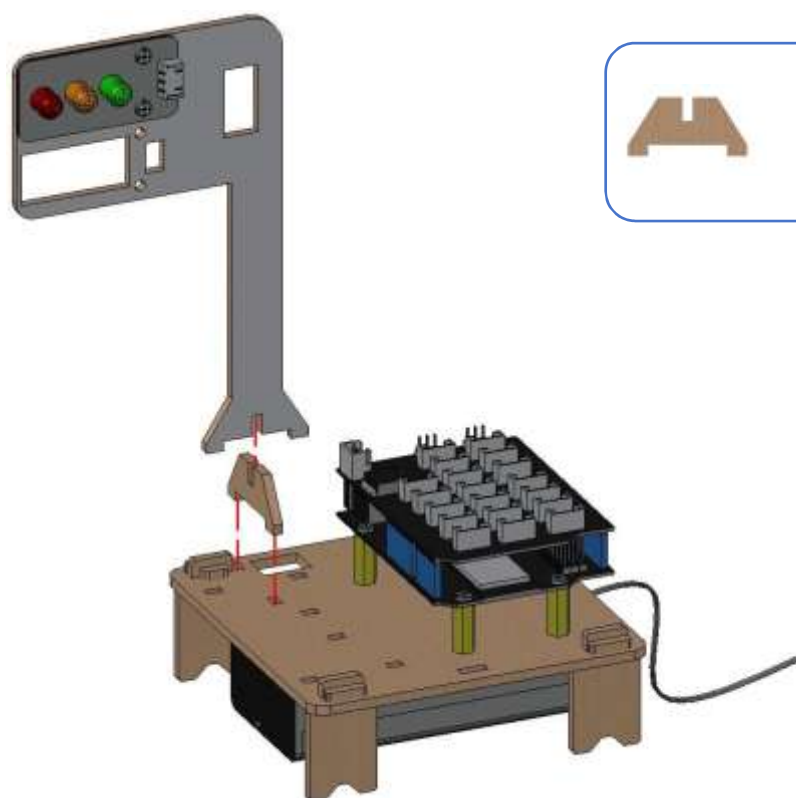


Krok 1: Instalace modulu semaforu.

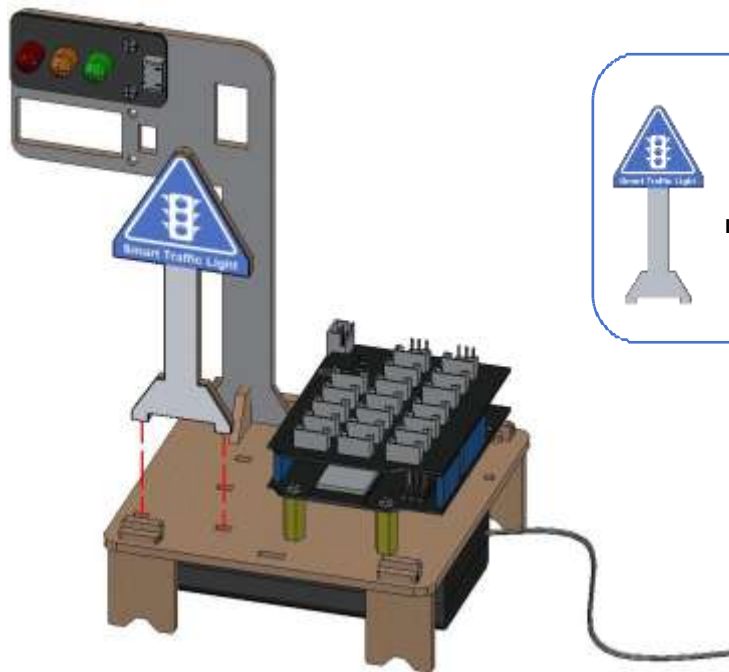
Poznámka: Před instalací modulu semaforu se ujistěte, že jste předem připojili 4pinový senzorový kabel.



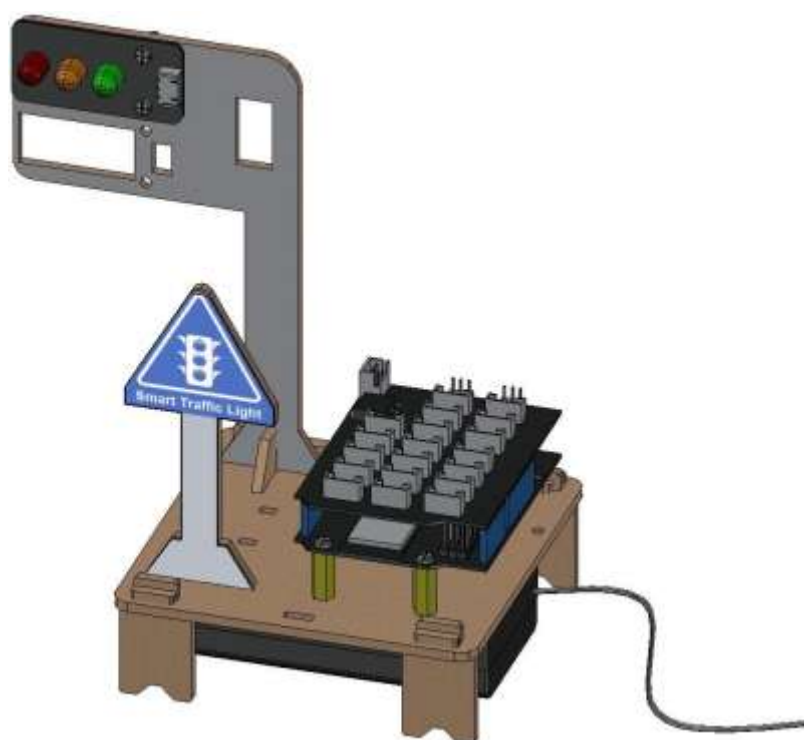
Krok 2: Upevněte modul semaforu na základnu.



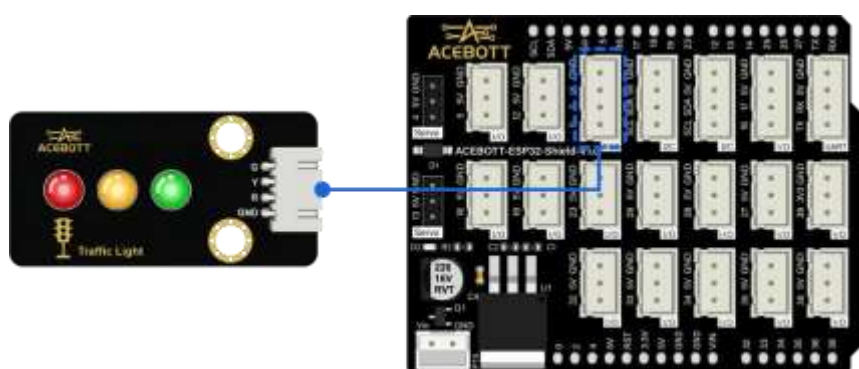
Krok 3: Instalace stojícího označení na základnu.



Intelligent traffic light signagex 1



Krok 4: Zapojení.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
Traffic light module	G	0
	Y	2
	R	14
	GND	GND

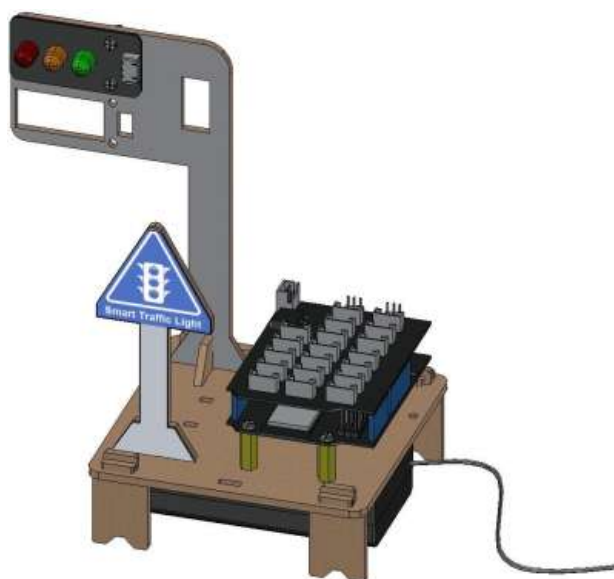
Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

05 Semaforey s odpočítáváním

Upozornění: Tento projekt bude vycházet ze struktury

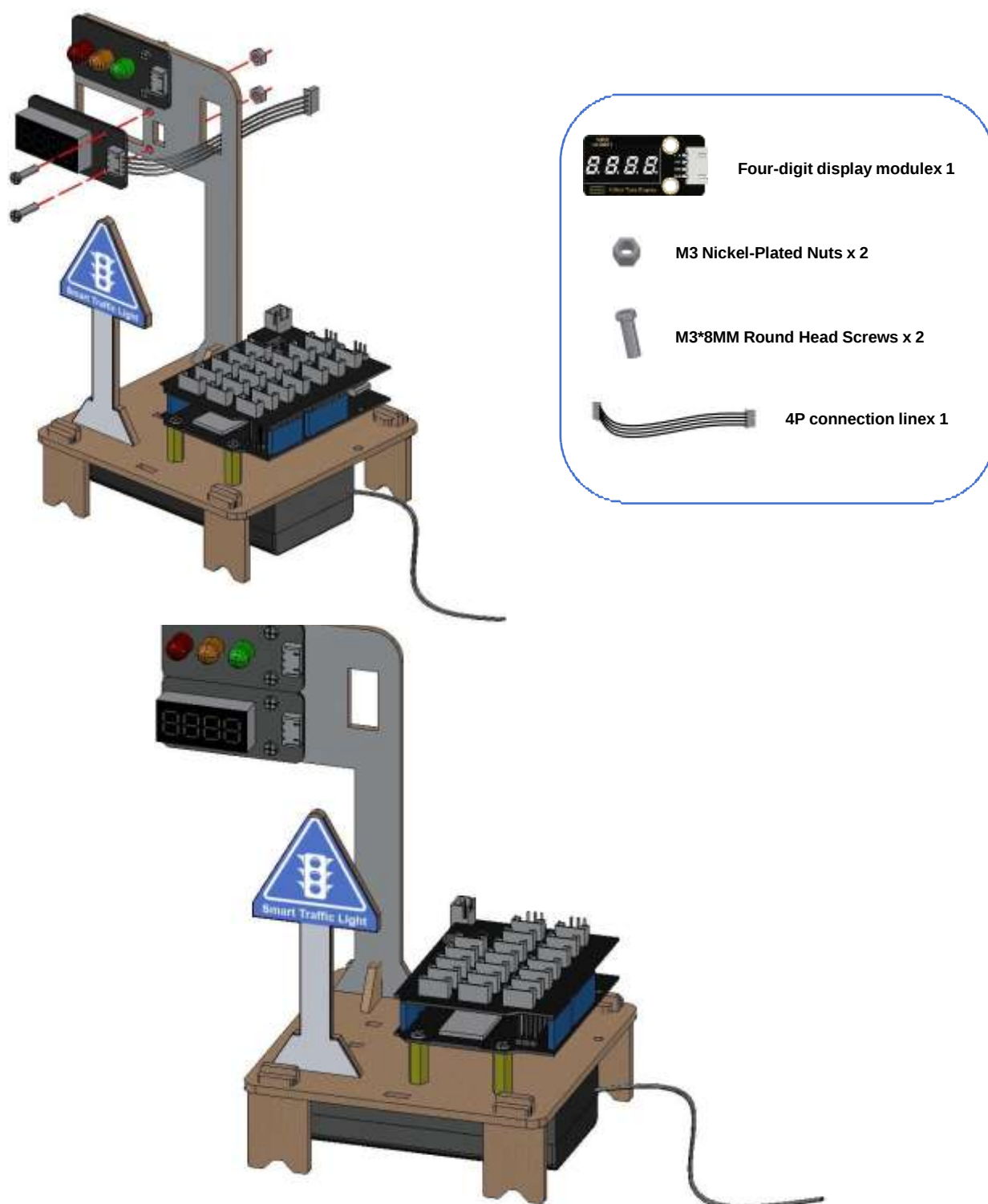
„Základních semaforů“ (viz obrázek níže). Konkrétní kroky konstrukce zde nebudou zobrazeny. Přejdeme přímo k instalaci



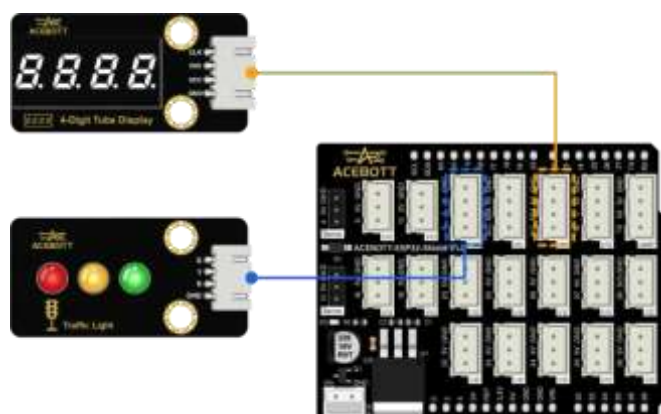
čtyřmístného displeje.

Krok 1: Instalace modulu čtyřmístného displeje.

Poznámka: Před instalací modulu čtyřmístného displeje se ujistěte, že jste předem připojili 4pinový konektor senzoru.



Krok 2: Zapojení vodičů.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
Traffic light module	G	0
	Y	2
	R	14
	GND	GND
Four-digit display module	CLK	SCL
	DIN	SDA
	VCC	5V
	GND	GND

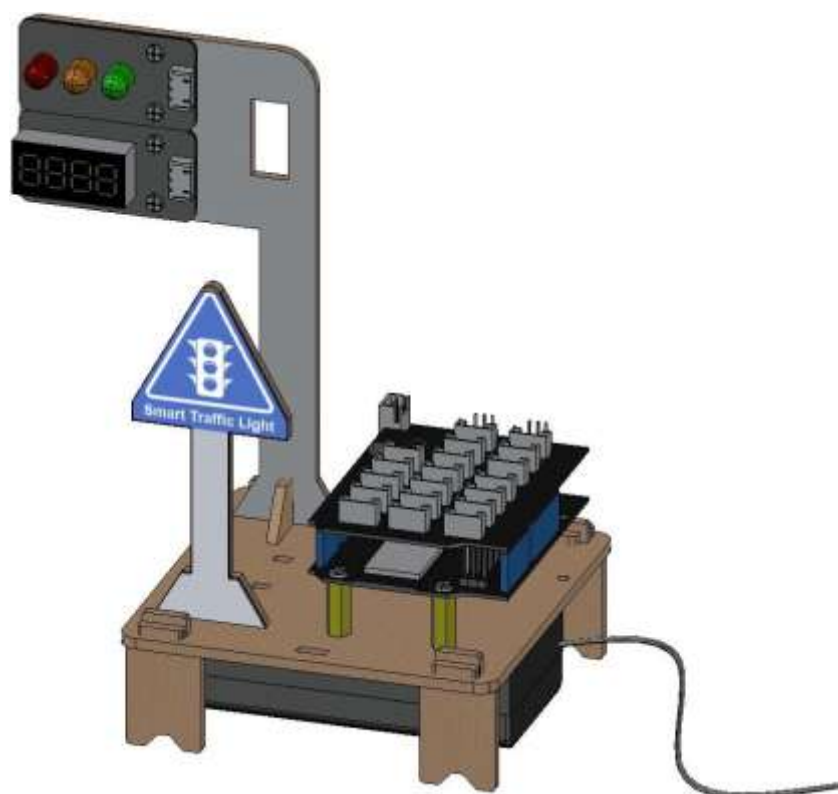
Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

06 Inteligentní semaforey

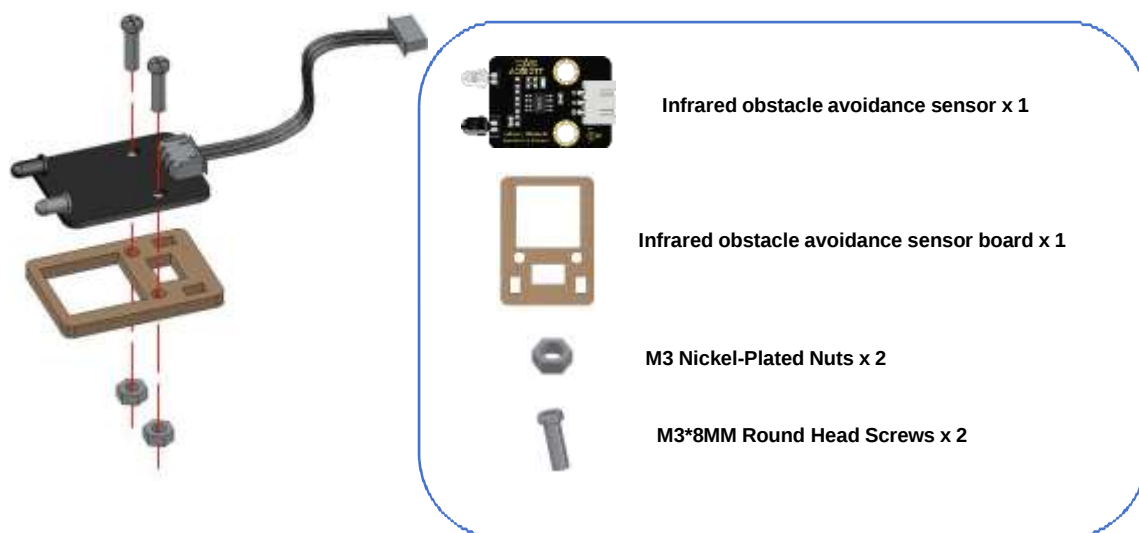
Upozornění: Tento projekt bude vycházet ze struktury

„Semaforů s odpočítáváním“ (viz obrázek níže). Konkrétní kroky konstrukce zde nebudou zobrazeny. Přejdeme přímo k sestavení infračerveného modulu pro vyhýbání se překážkám.

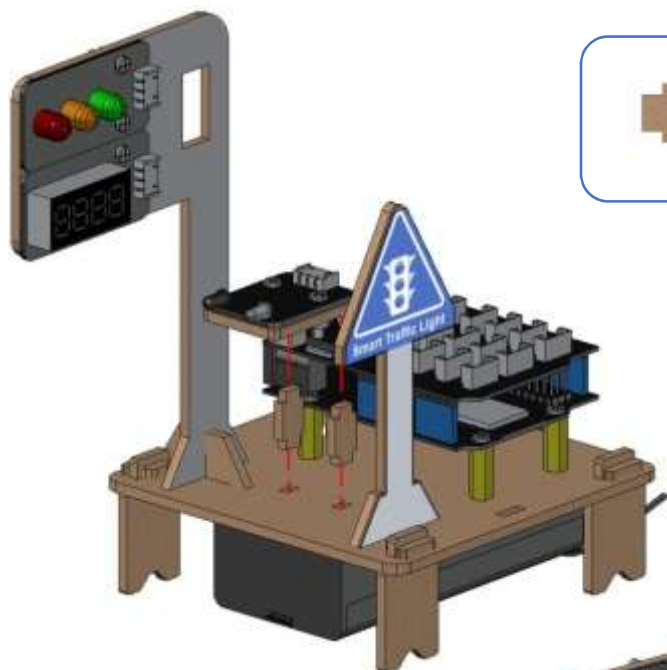


Krok 1: Instalace infračerveného senzoru pro vyhýbání se překážkám.

Poznámka: Před upevněním infračerveného senzoru pro vyhýbání se překážkám připojte předem 3pinový kabel senzoru.



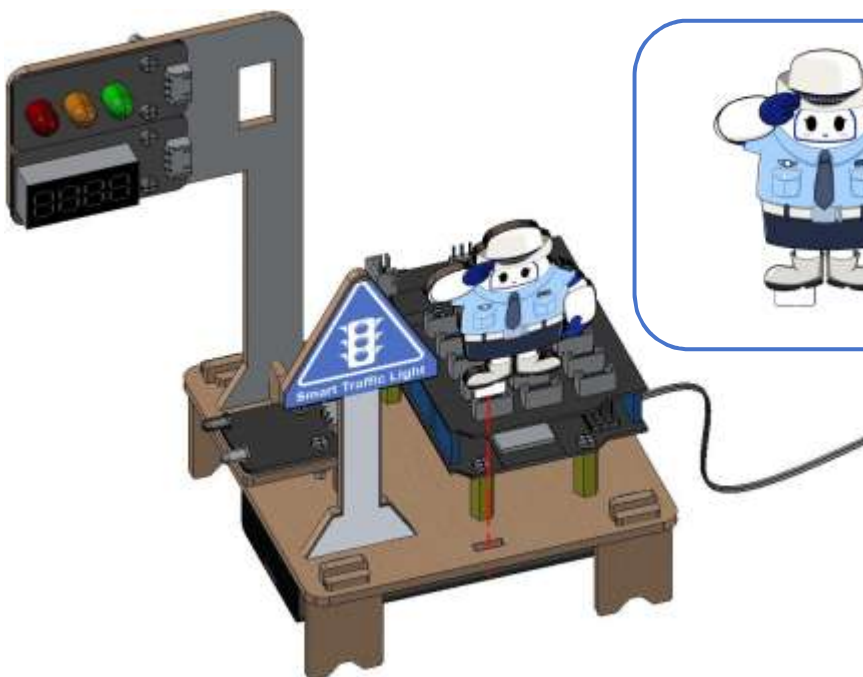
Krok 2: Upevněte infračervený senzor pro vyhýbání se překážkám.



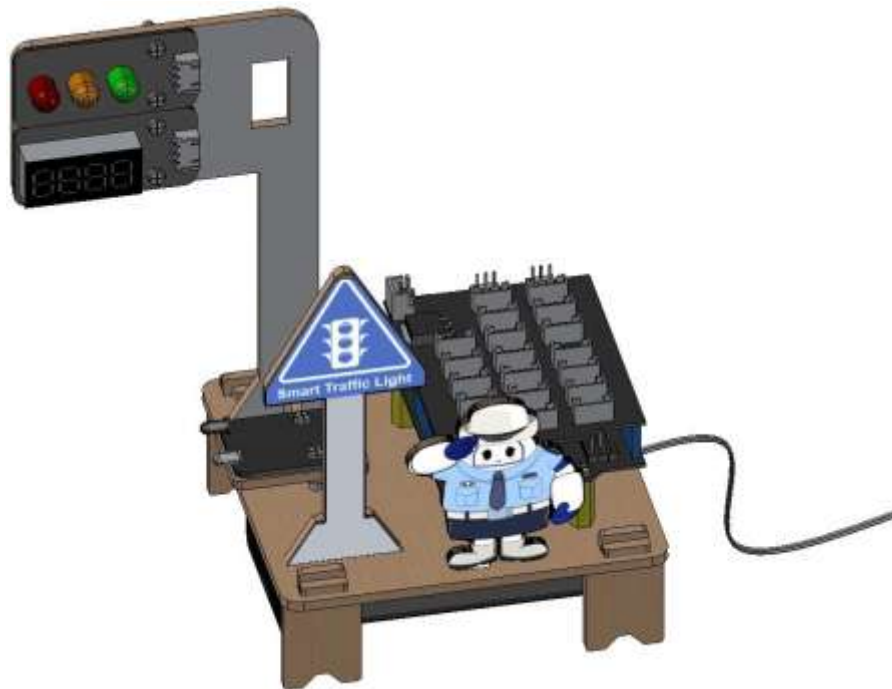
Secure the bracket boardx 2



Krok 3: Instalace LUMI.

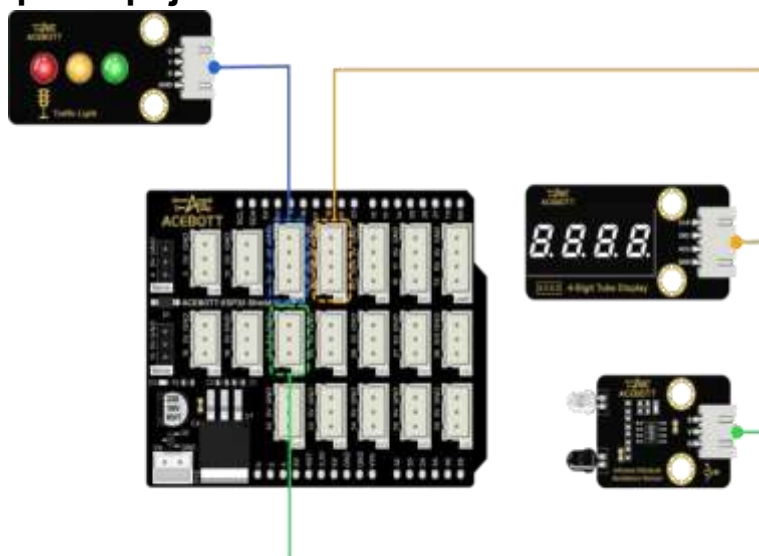


LUNA
standing
cardx 1



Krok 4: Zapojení vodičů.

Pokyny pro zapojení:



Název modulu	Pin modulu	Pin ESP32
Traffic light module	G	0
	Y	2
	R	14
	GND	GND
Four-digit display module	CLK	SCL
	DIN	SDA
	VCC	5V
	GND	GND
Infrared obstacle avoidance sensor	S	Pin23
	V	5V
	G	GND

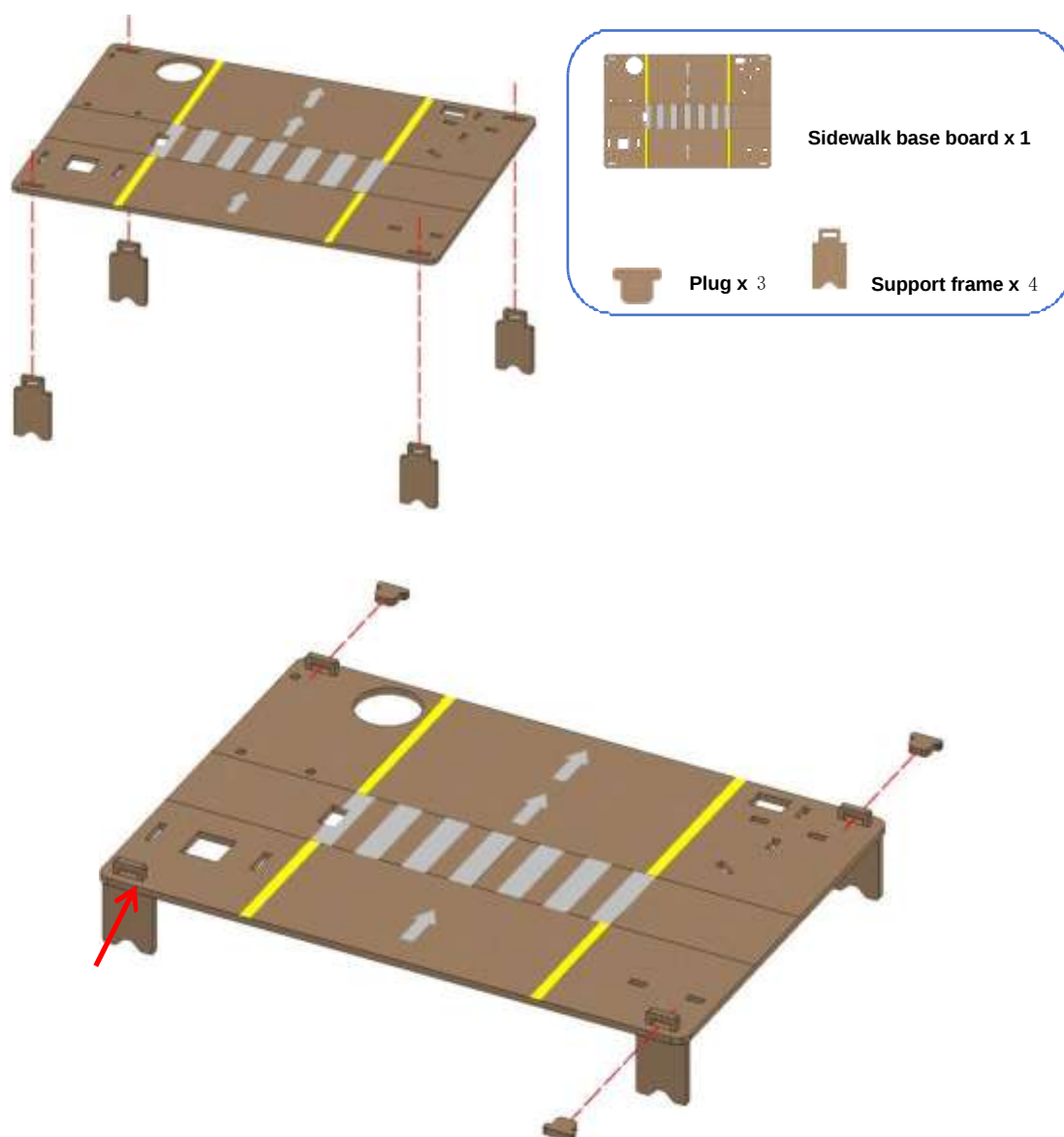
Poznámka: Při připojování modulu k ovládací desce ESP32

důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

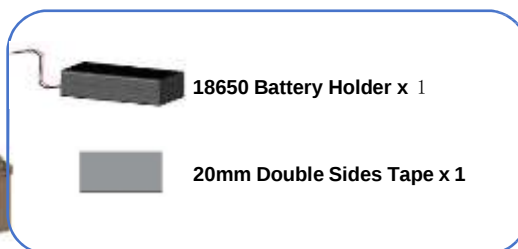
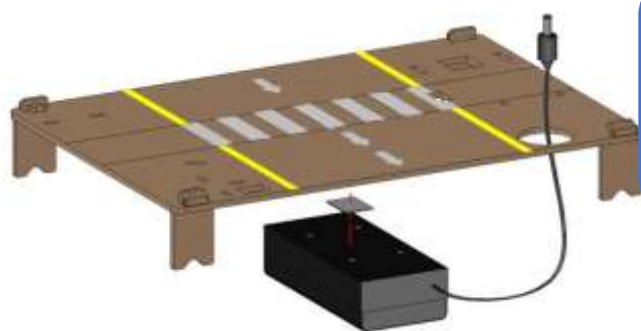
07 Inteligentní bezpečnostní ochrana pro přechod

Krok 1: Sestavte základnu.



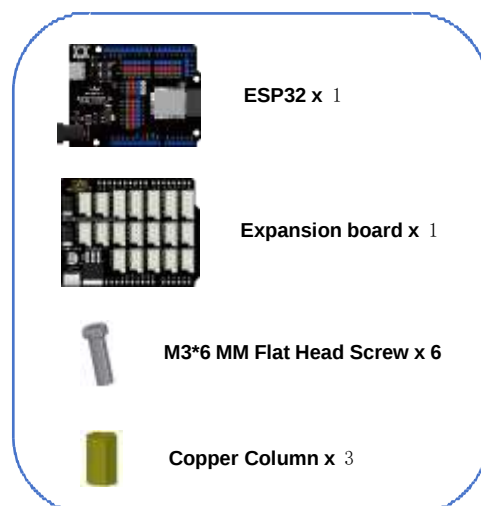
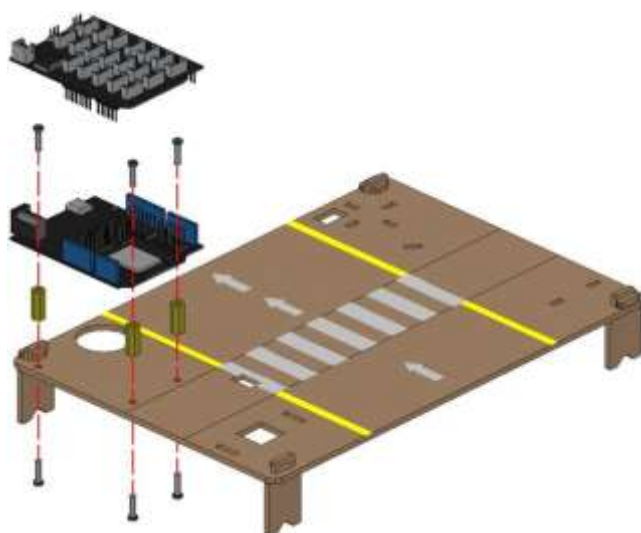
Poznámka: Zde se zástrčka namontuje později.

Krok 2: Instalace držáku baterie.



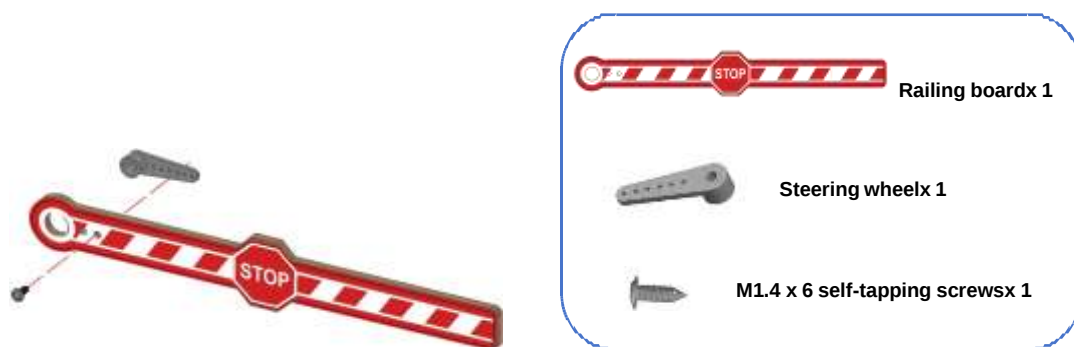
Poznámka: Použijte oboustrannou lepicí pásku k připevnění držáku baterie na spodní stranu základny a nezakrývejte elektrické vodiče a otvory pro piny na základně.

Krok 3: Instalace řídicí desky ESP32 a expanzní desky.

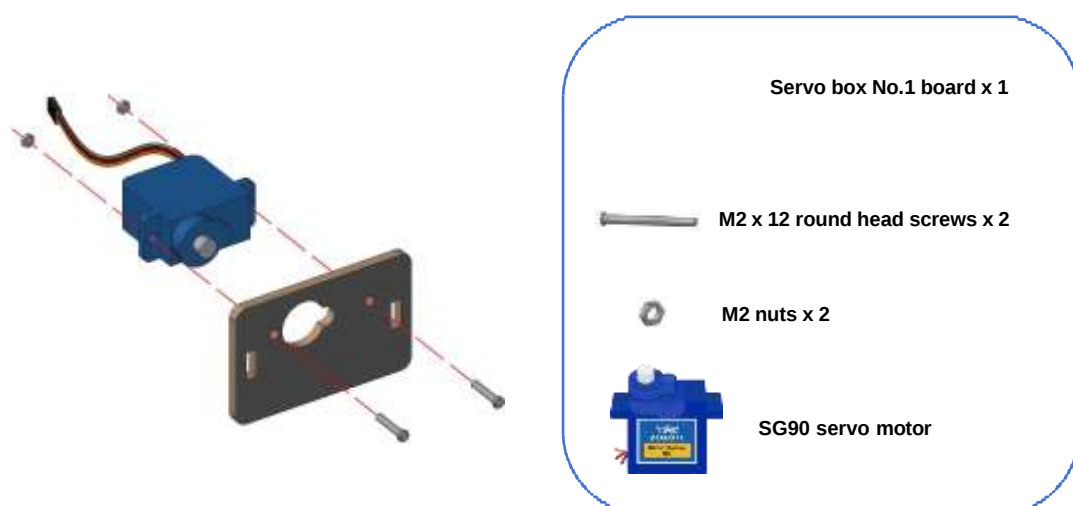


Poznámka: Nejprve připevněte měděný sloupek na základnu, poté nainstalujte řídicí desku ESP32 na měděný sloupek a nakonec nainstalujte expanzní desku. Při instalaci expanzní desky se ujistěte, že otvory jsou správně zarovnány s ESP32 před jejím upevněním.

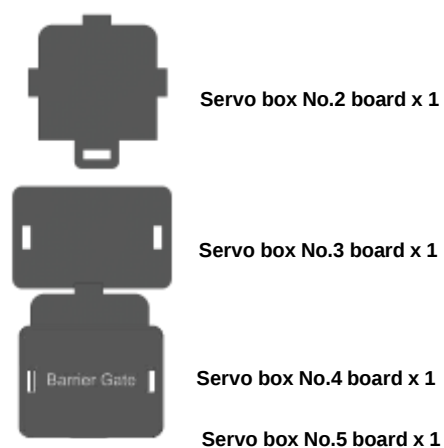
Krok 4: Instalace zábradlí.

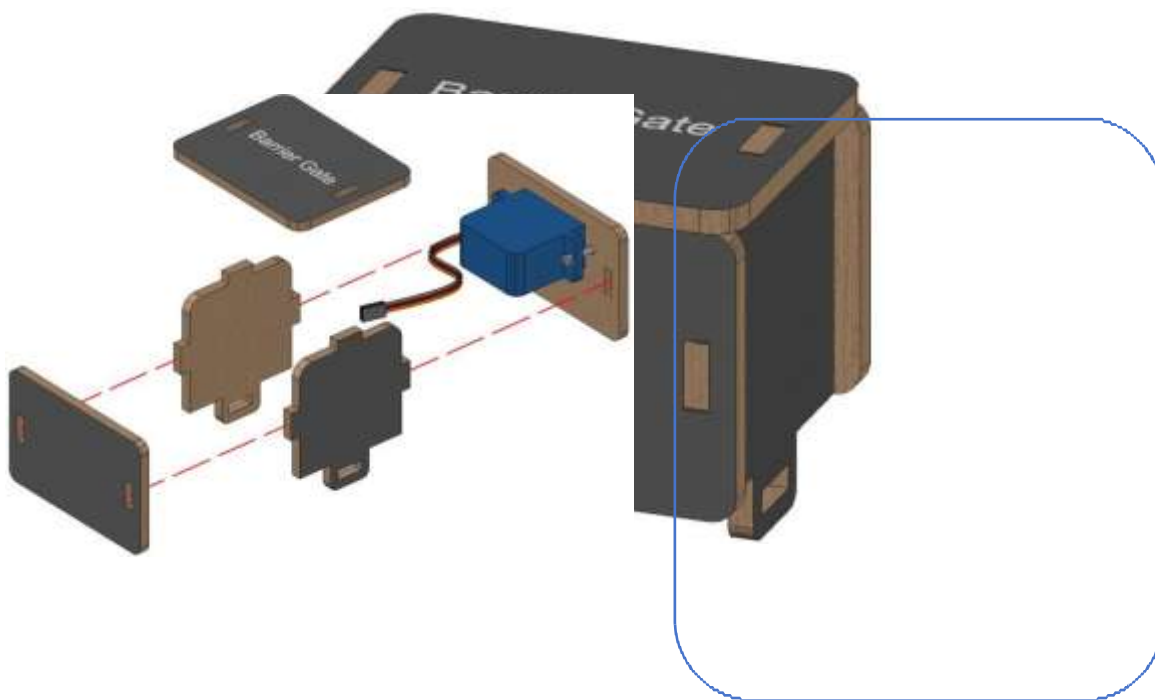


Krok 5: Instalace servomotoru.



Poznámka: Sestavte desky servo boxu z dýhových desek v číselném pořadí a nakonec zakryjte desku č. 5.





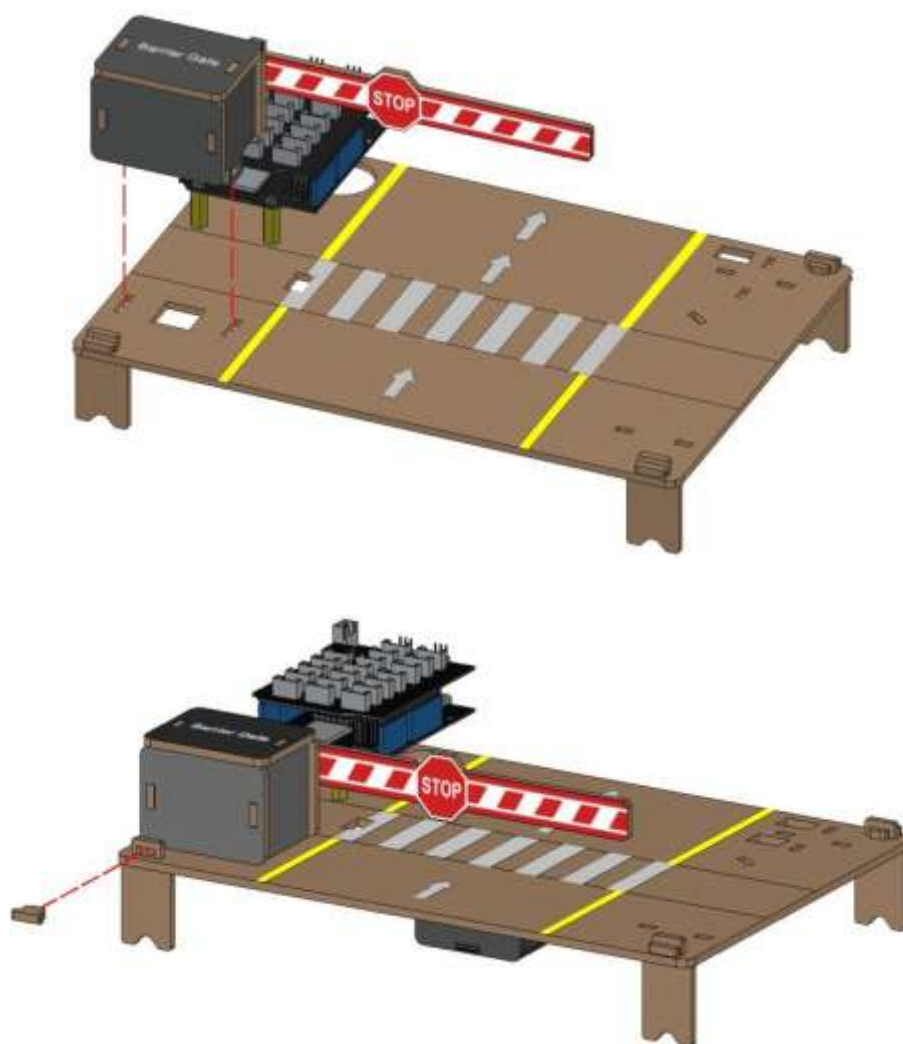
Krok 6: Upevněte konstrukci zábradlí.

Poznámka: Před instalací tohoto kola na zábradlí nastavte servomotor na pozici 90 stupňů. Po nastavení na pozici 90 stupňů nezapínejte napájení servomotoru. Poté nainstalujte zábradlí podle následujícího diagramu.

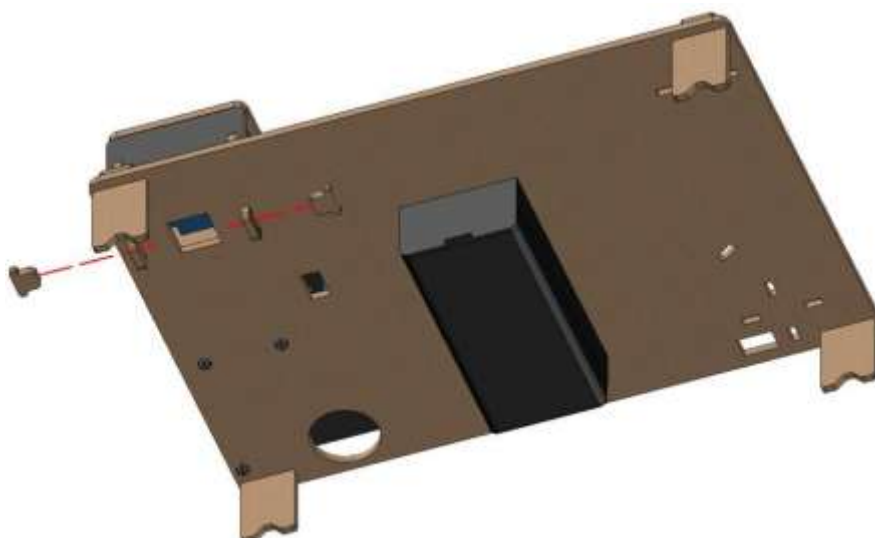
[Klikněte zde pro přístup k programu pro nastavení servomotoru.](#)



Krok 7: Připevněte box servomotoru na základnu.

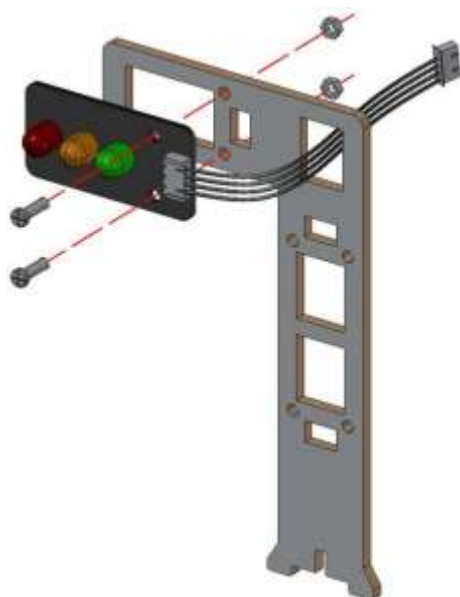


Upozornění: V tomto kroku se montuje poslední zástrčka podstavce.

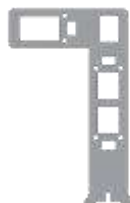


Krok 8: Nainstalujte modul semaforu.

Před instalací modulu semaforu se ujistěte, že jste předem připojili 4-pinový konektor senzoru.



Traffic light module x 1



Traffic light bracket board x 1



M3 Nickel-Plated Nuts x 2

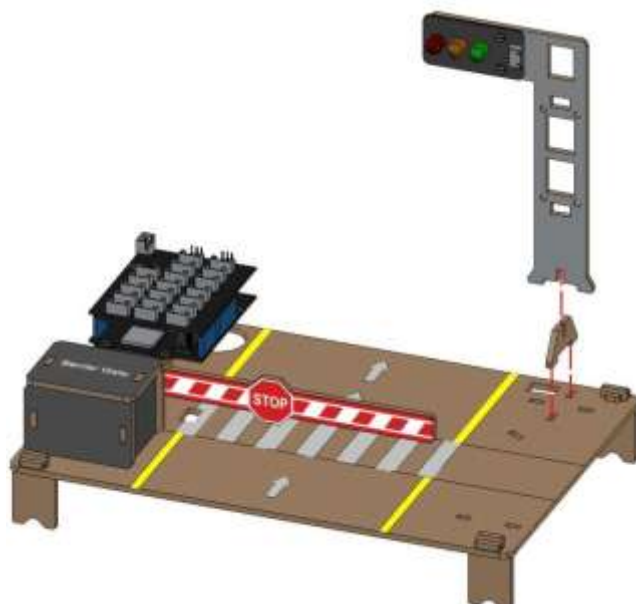


M3*8 MM Round Head Screws x 2



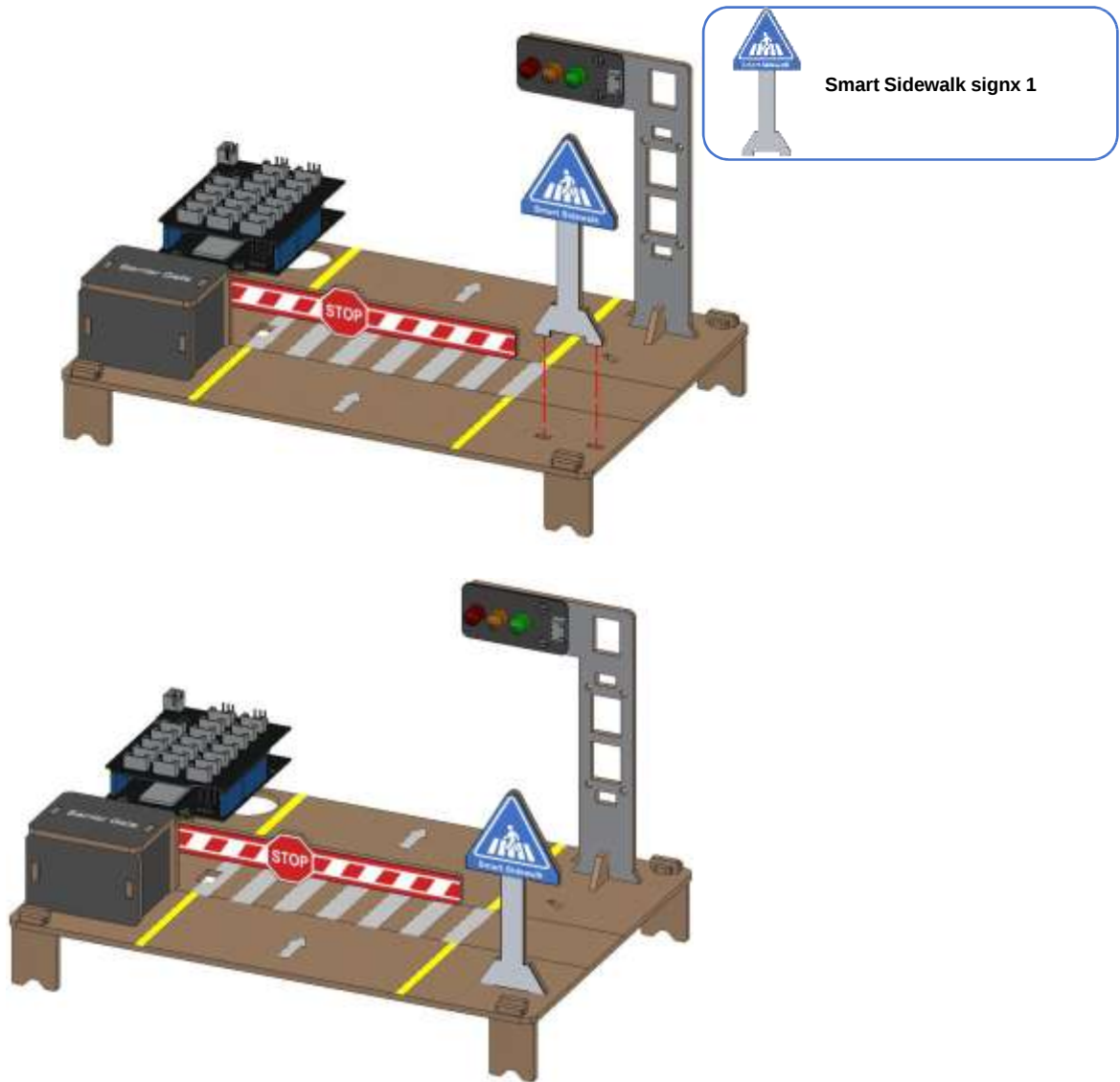
4P connection line x 1

Krok 9: Upevněte modul semaforu na základnu.

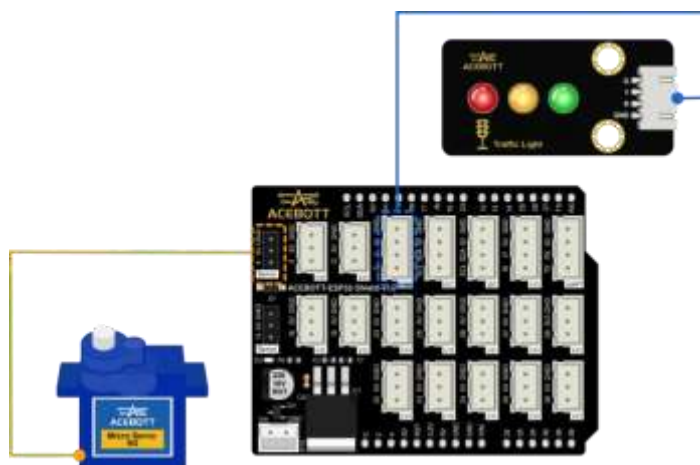


Bracket board.x 1

Krok 10: Nainstalujte značení na základnu.



Krok 11: Kabeláž.



Pokyny pro zapojení:

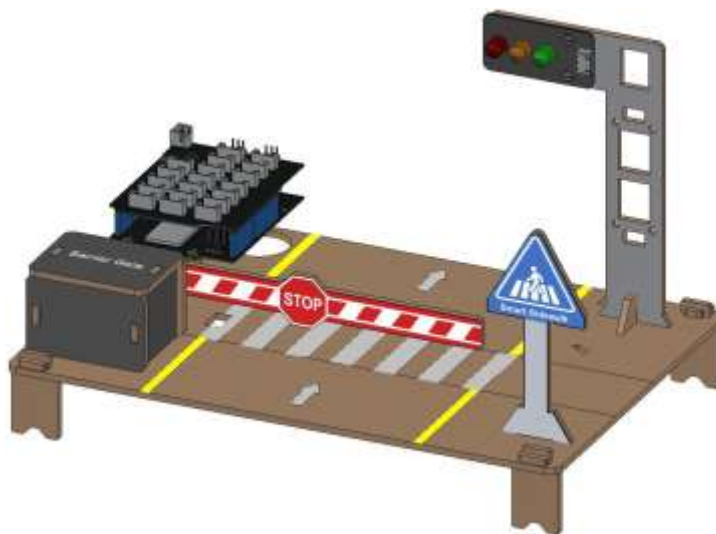
Název modulu	Pin modulu	Pin ESP32
Traffic light module	G	0
	Y	2
	R	14
	GND	GND
Servo motor	S	4
	V	5V
	G	GND

Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

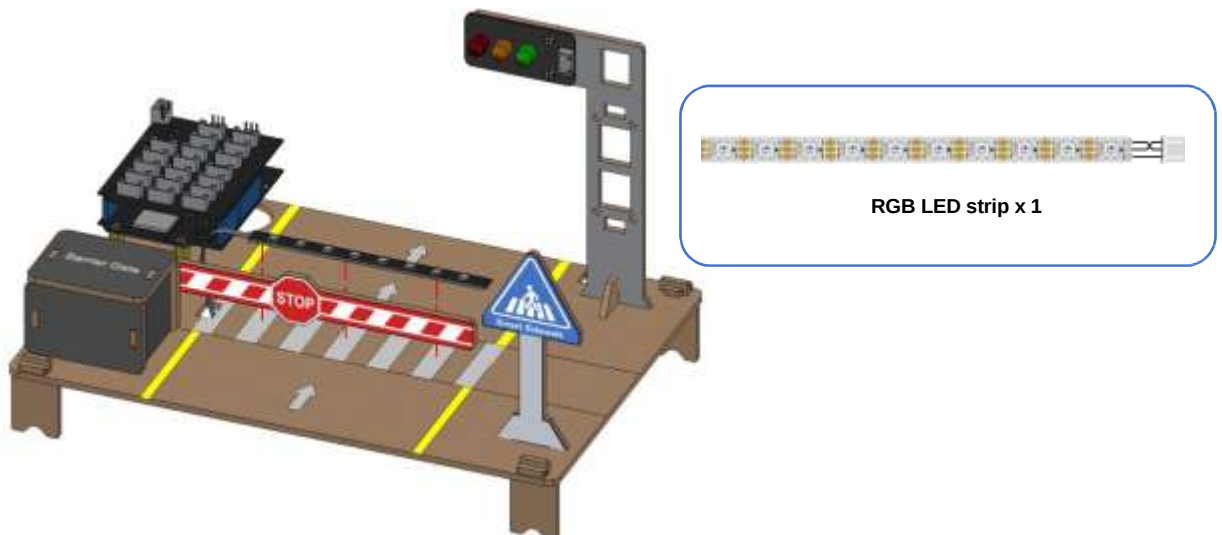
08 Dopravní světla na zemi

Teplé upozornění: Tento projekt bude následovat strukturu "Inteligentní bezpečnostní ochrany pro chodce" (jak je znázorněno na obrázku níže). Konkrétní konstrukční kroky zde nebudou zobrazeny. Přejdeme přímo k sestavení RGB LED pásku.



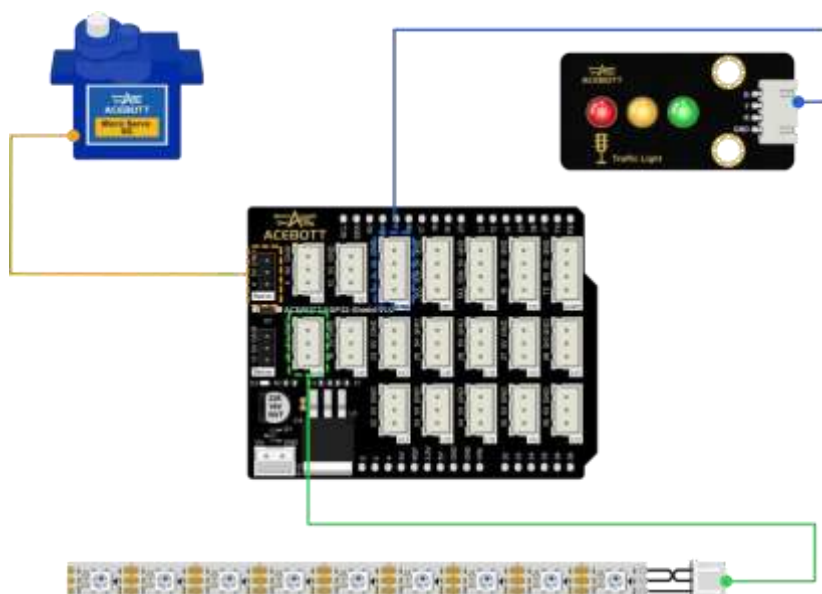
Krok 1: Nainstalujte RGB LED pásek.

Upozornění: Odstraňte ochrannou fólii na zadní straně LED pásku a poté jej přilepte na chodník pro chodce.





Krok 2: Drátování.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
Traffic light module	G	0
	Y	2
	R	14
	GND	GND
Servo motor	S	4
	V	5V
	G	GND
RGB LED strip	DIN	18
	5V	5V
	GND	GND

Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

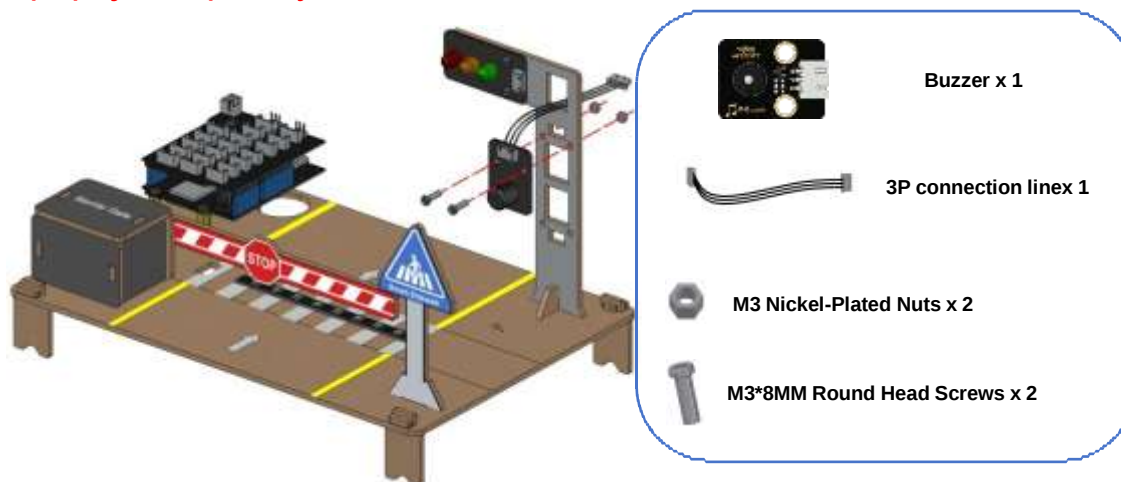
09 Bezpečnostní připomenutí pro přechod

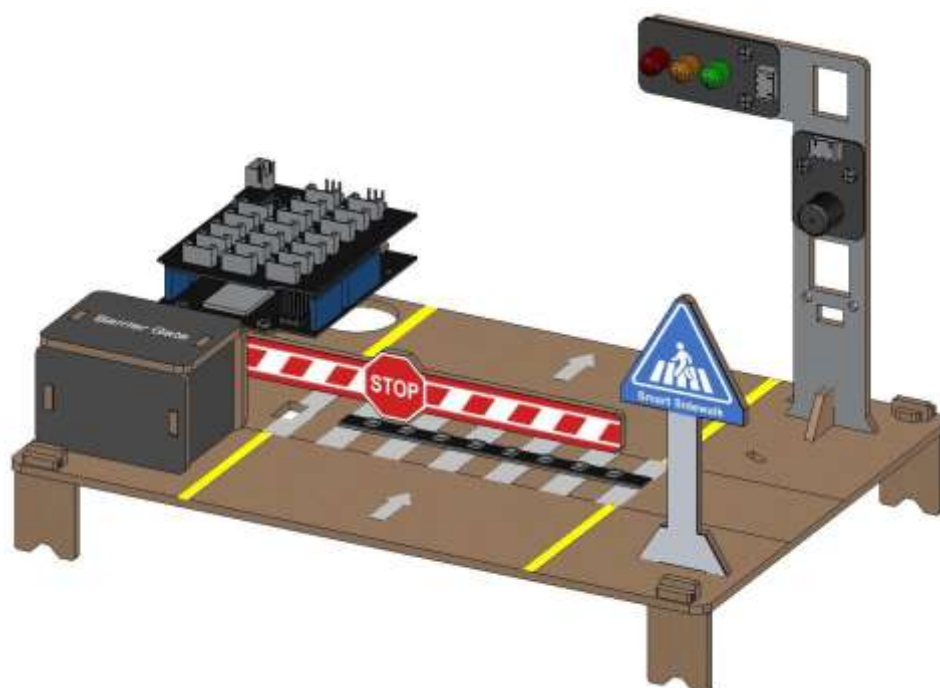
Teplý pokyn: Tento projekt bude následovat strukturu "Chodníkových světel pro bezpečnost přechodu" (jak je zobrazeno na obrázku níže). Konkrétní kroky konstrukce zde nebudou zobrazeny. Přejdeme přímo k výstavbě bzučáku.



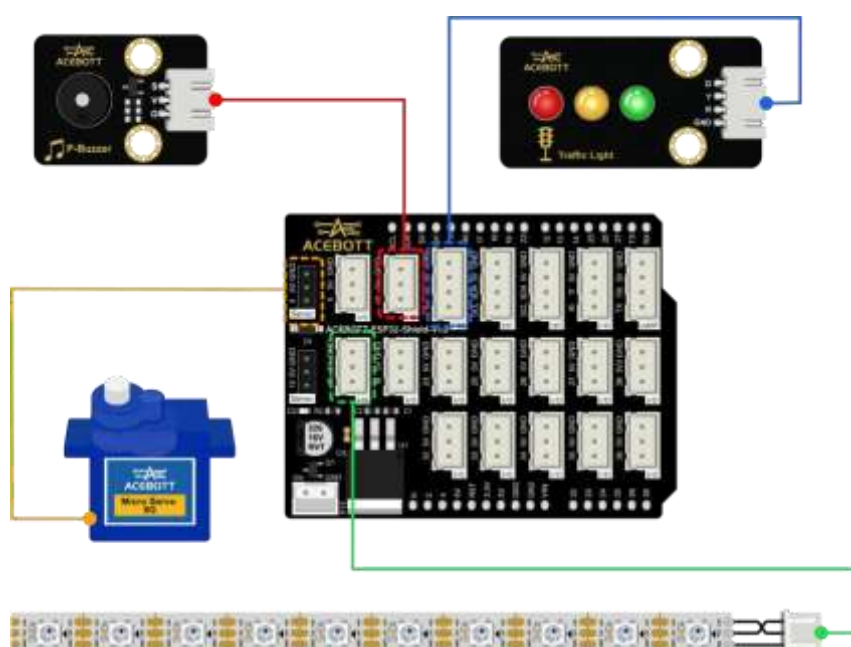
Krok 1: Nainstalujte bzučák.

Upozornění: Před instalací bzučáku se ujistěte, že předem připojíte 3-pinový konektor senzoru.





Krok 2: Kabeláž.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
Traffic light module	G	0
	Y	2
	R	14
	GND	GND
Servo motor	S	4
	V	5V
	G	GND
RGB LED strip	DIN	18
	5V	5V
	GND	GND
Buzzer	S	12
	V	5V
	G	GND

Poznámka:Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

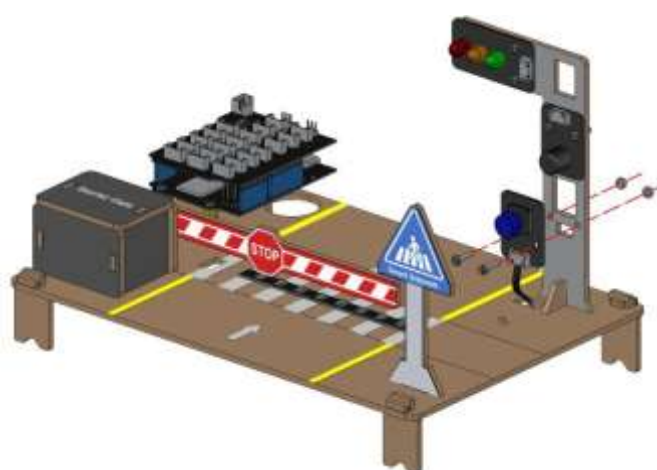
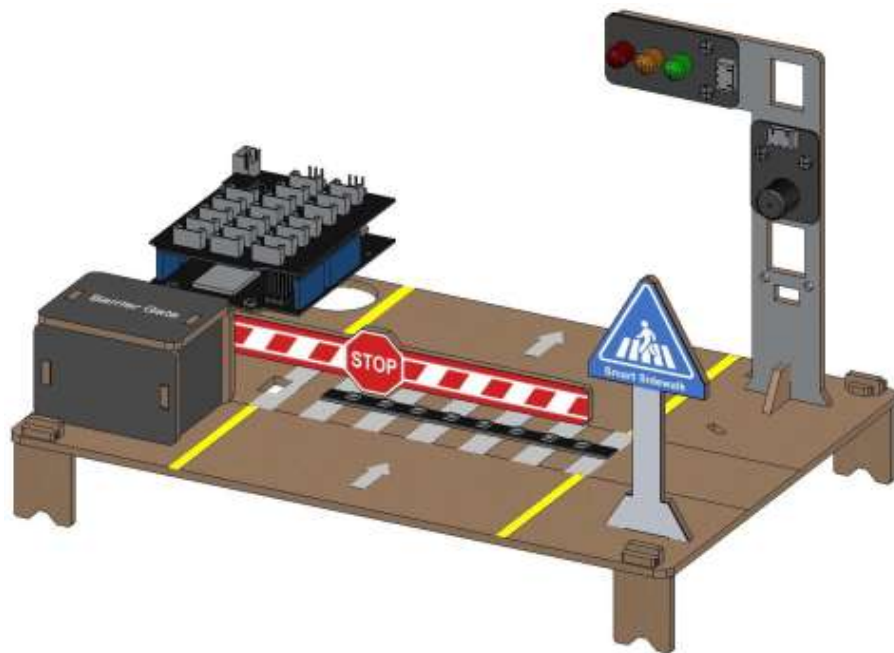
10 Tlačítkové semaforey

Teplé upozornění: Tento projekt bude vycházet ze struktury

„Připomenutí bezpečnosti chodců“ (viz obrázek níže).

Konkrétní stavební kroky zde nebudou zobrazeny.

Přistoupíme přímo k montáži tlačítkového modulu.



Button module x 1



M3 Nickel-Plated Nuts x 2

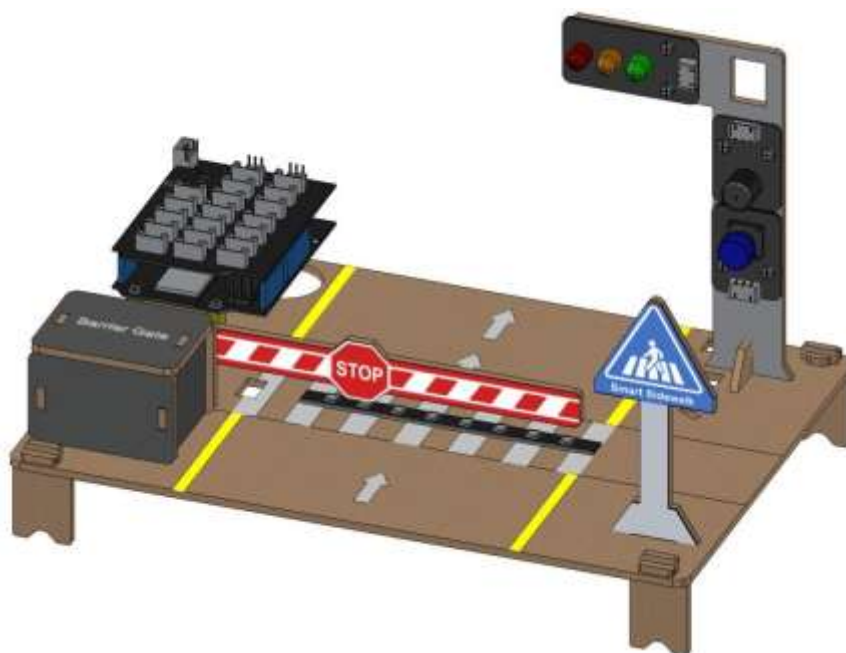


M3*8MM Round Head Screws x 2



3P connection linex 1

Krok 1: Sestavte tlačítkový modul.



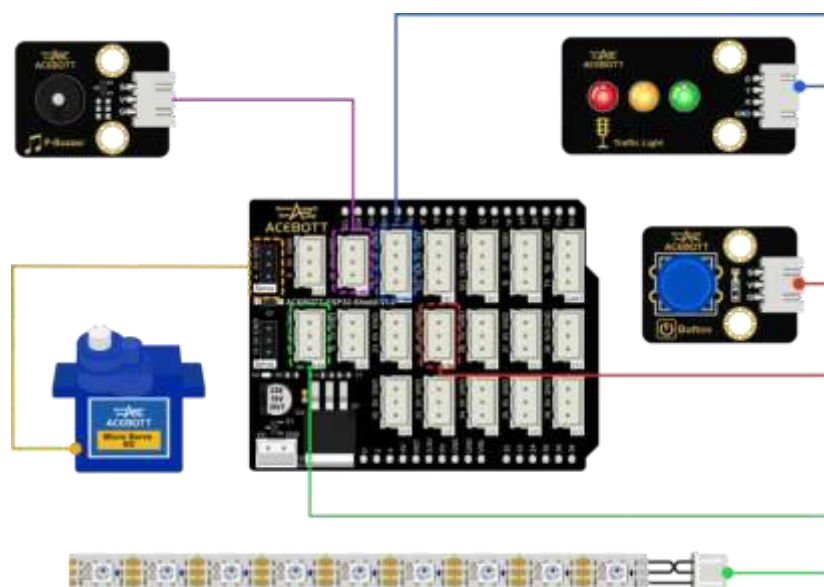
Krok 2: Nainstalujte LUMI.



LUNA standing cardx 1



Krok 3: Zapojení.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
	G	0

Traffic light module	Y	2
	R	14
	GND	GND
Servo motor	S	4
	V	5V

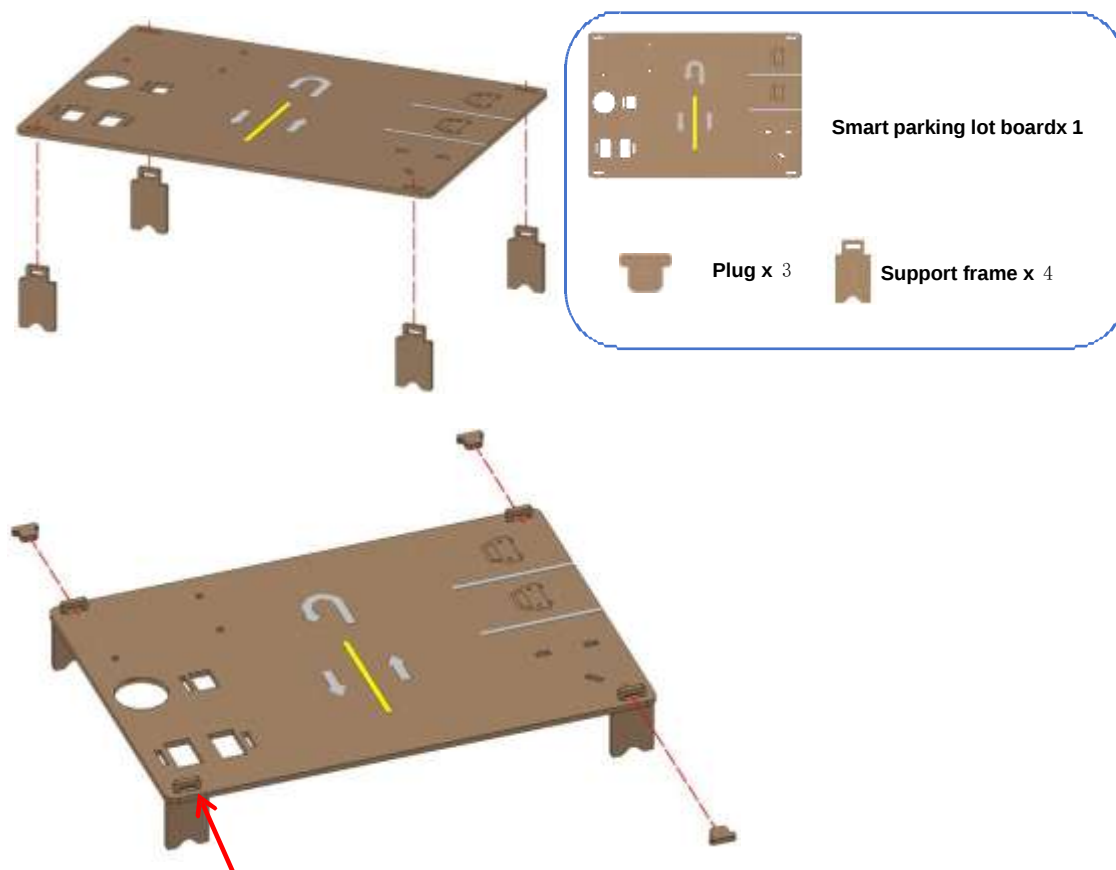
	G	GND
RGB LED strip	DIN	18
	5V	5V
	GND	GND
Button Module	S	25
	V	5V
	G	GND
Buzzer	S	12
	V	5V
	G	GND

Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

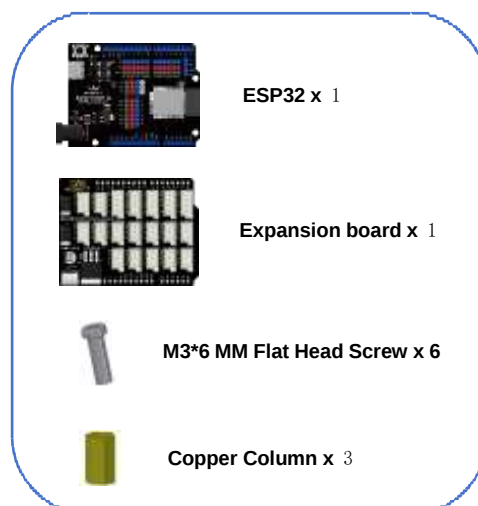
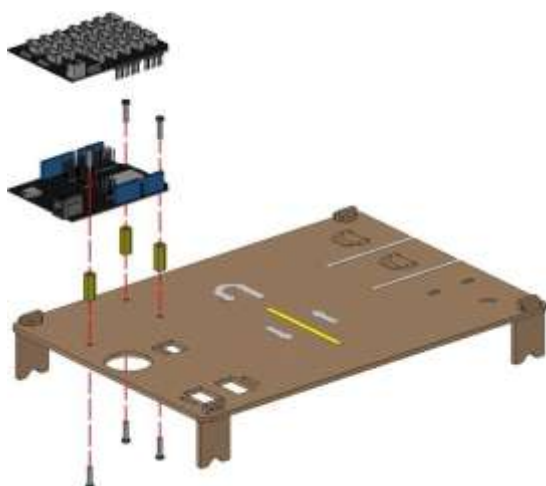
11 Inteligentní přístupová kontrola

Krok 1: Sestavte základnu.



Zde se zástrčka namontuje později.

Krok 2: Nainstalujte řídicí desku ESP32 a rozšiřující desku.

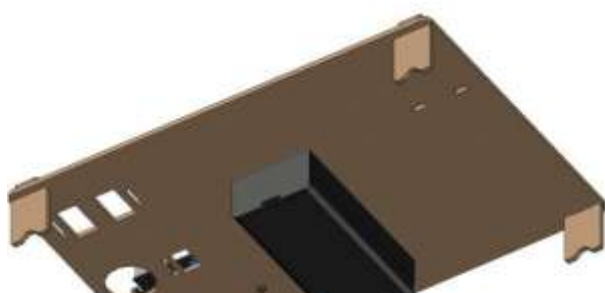


Upozornění: Nejprve upevněte měděný sloupek na základnu, poté na sloupek připevněte řídicí desku ESP32 a nakonec nainstalujte rozšiřující desku. Při instalaci rozšiřující desky dbejte na to, aby byly otvory zarovnány s deskou ESP32.

Krok 3: Instalace držáku baterie.



Upozornění: Použijte oboustrannou pásku k připevnění držáku baterie na spodní stranu základny a nezakrývejte dráty a otvory pro piny na základně.

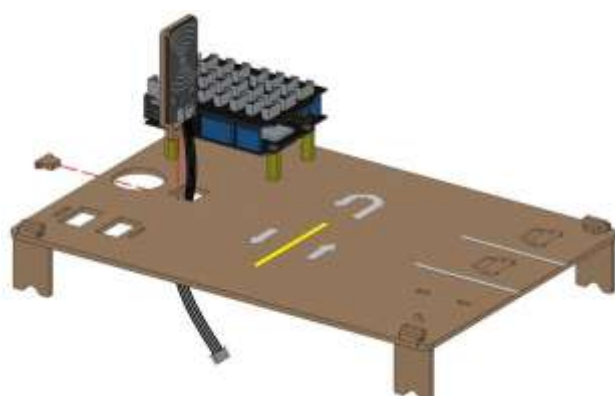


Krok 4: Sestavte RFID modul.



Krok 5: Nainstalujte RFID modul na základnu.

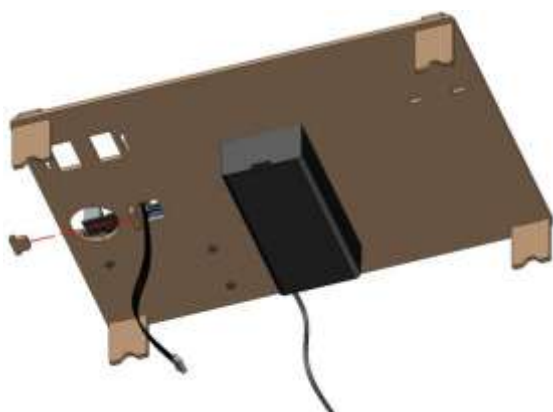
Poznámka: Před upevněním RFID modulu se ujistěte, že jste předem připojili 4pinový senzorový kabel.



4-pin sensor cable x 1



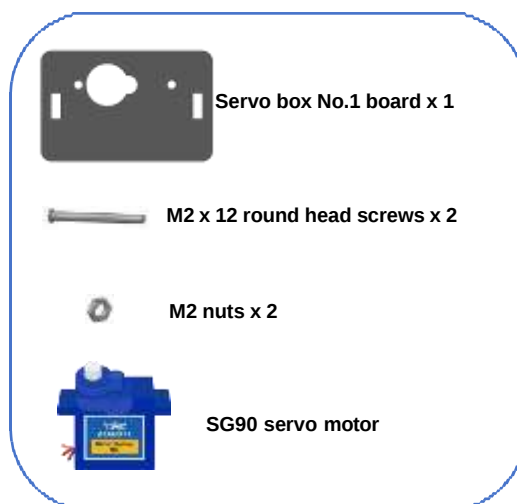
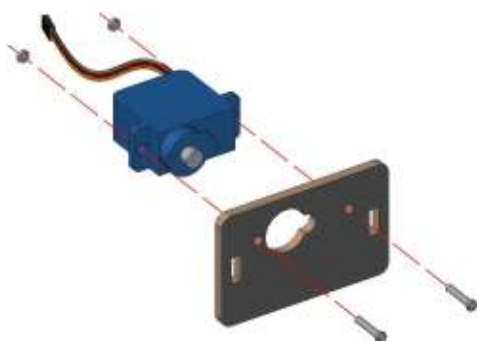
Plug x 1



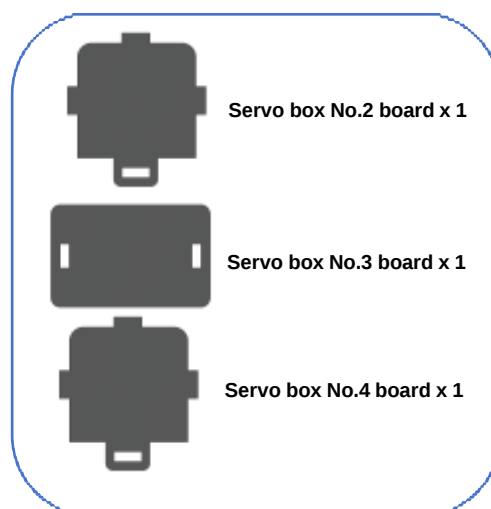
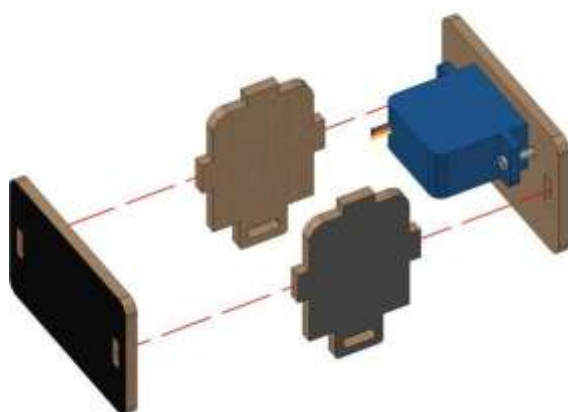
Krok 6: Nainstalujte zábradlí.



Krok 7: Nainstalujte servo motor.



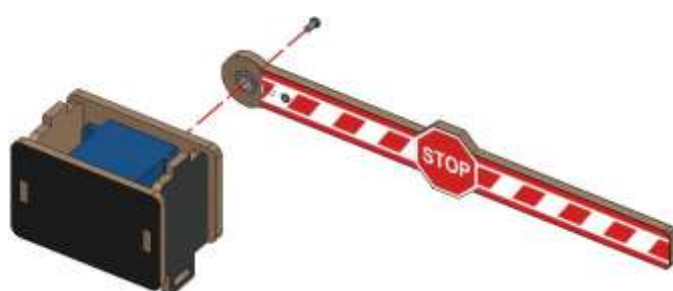
Poznámka: Dřevěné desky servo krabice se skládají dohromady podle pořadových čísel.



Krok 8: Upevněte konstrukci zábradlí.

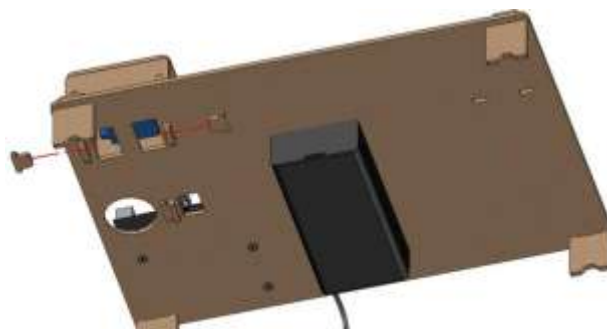
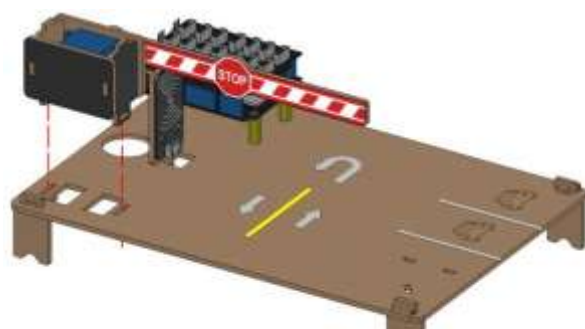
Poznámka: Před instalací tohoto kola na zábradlí nejprve nastavte servo na 90stupňovou pozici. Po nastavení na 90stupňovou pozici nezapínejte servo. Poté nainstalujte zábradlí podle níže uvedeného diagramu.

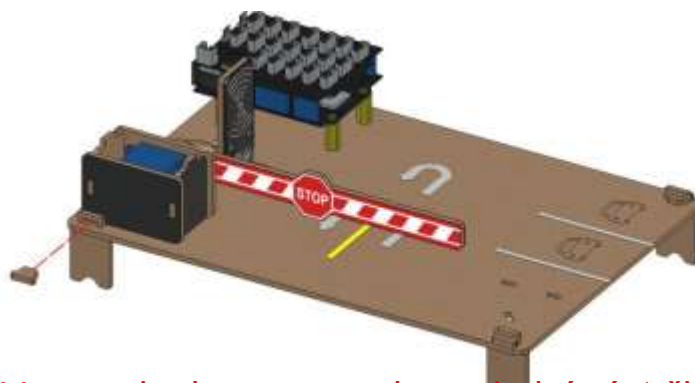
[Klikněte zde pro získání programu pro ladění serva.](#)



Servo motor comes with
built-in mounting screw x 1

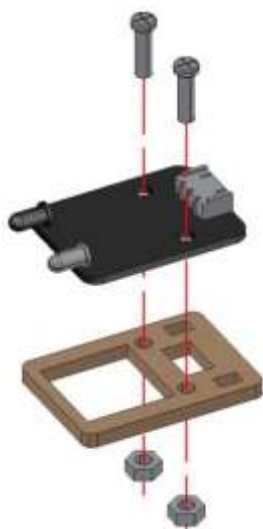
Krok 9: Nainstalujte servo strukturu.





Upozornění: V tomto kroku se montuje poslední zástrčka podstavce.

Krok 10: Nainstalujte senzor pro detekci překážek pomocí infračerveného záření.



Infrared obstacle avoidance sensor x 1



Infrared obstacle avoidance sensor board x 1

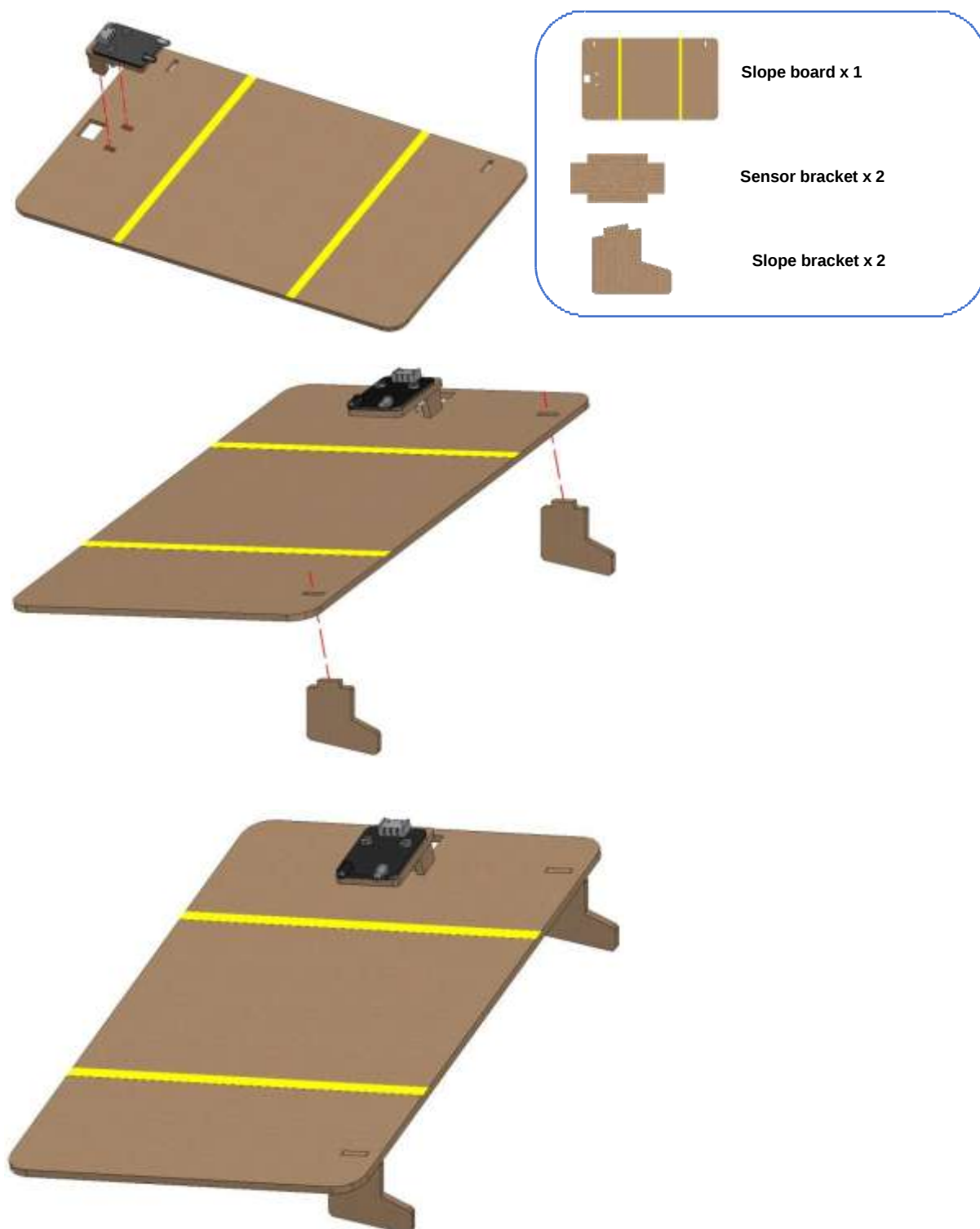


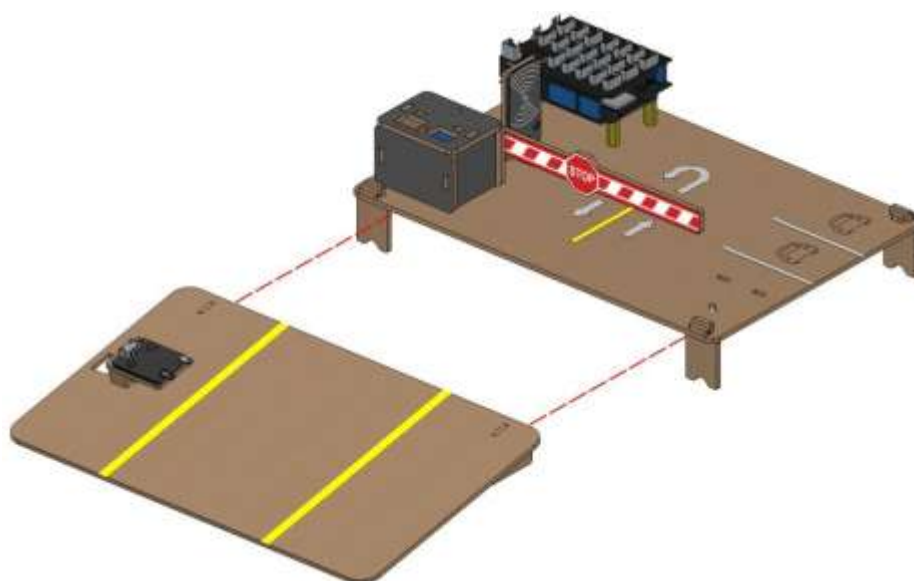
M3 Nickel-Plated Nuts x 2



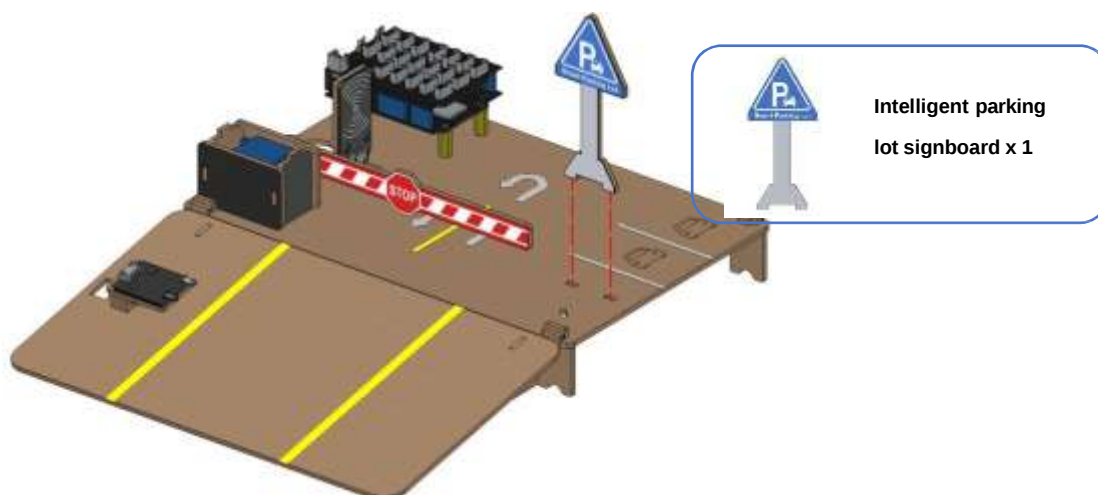
M3*8MM Round Head Screws x 2

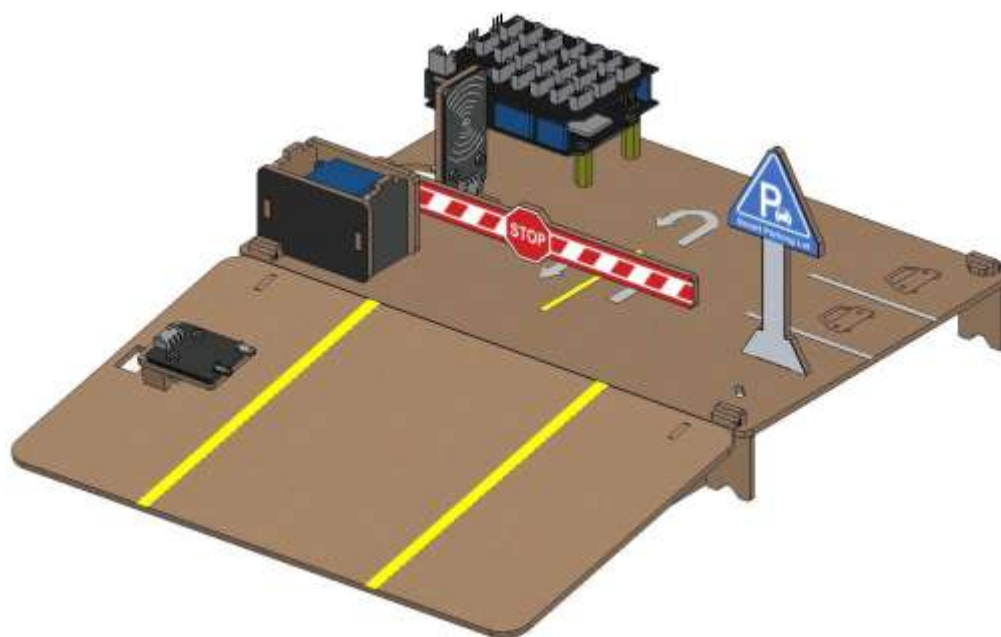
Krok 11: Upevněte senzor pro detekci překážek na skládací desku.



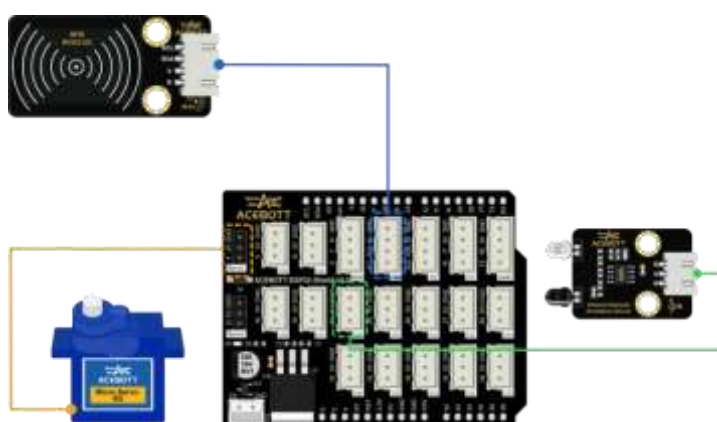


Krok 12: Nainstalujte inteligentní parkovací značku na základnu.





Krok 13: Kabeláž.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
RFID module	SCL	SCL
	SDA	SDA
	VCC	5V
	GND	GND
Servo motor	S	4
	V	5V
	G	GND

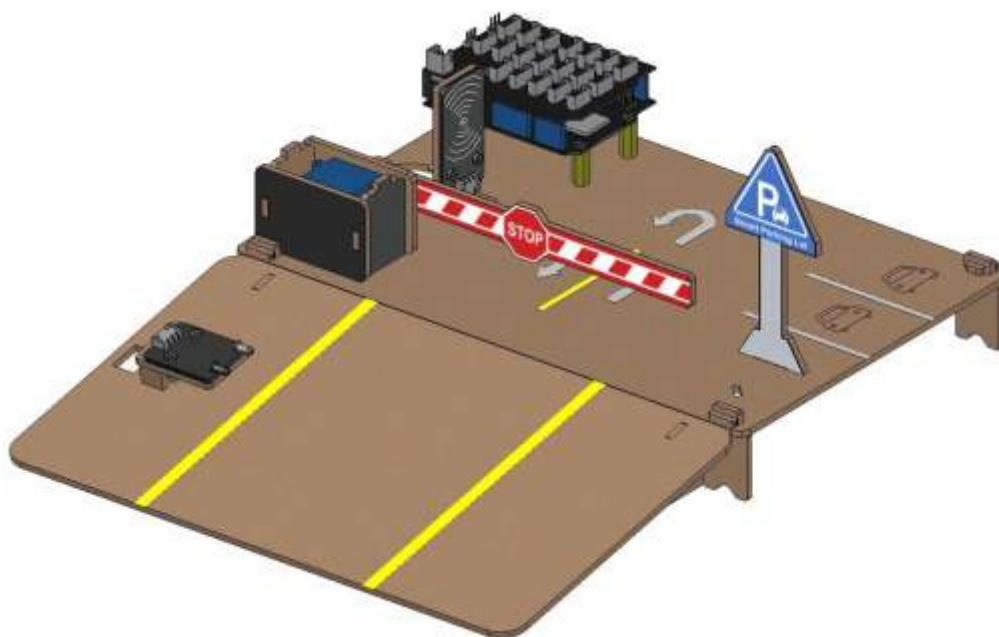
Infrared obstacle avoidance sensor	S	23
	V	5V
	G	GND

Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

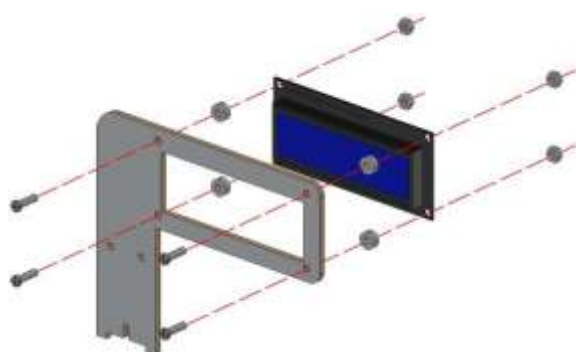
[Klikněte zde pro otevření programu pro tento projekt.](#)

12 Zobrazení parkovacích míst

Teplé upozornění: Tento projekt bude následovat strukturu „Inteligentní přístupové kontroly“ (jak je znázorněno na obrázku níže). Konkrétní kroky stavby zde nebudou zobrazeny. Pokračujeme přímo k instalaci LCD displeje.



Krok 1: Nainstalujte LCD displej.



LCD1602 module x 1



M3*3 gasket x 4



M3 Nickel-Plated Nuts x 4

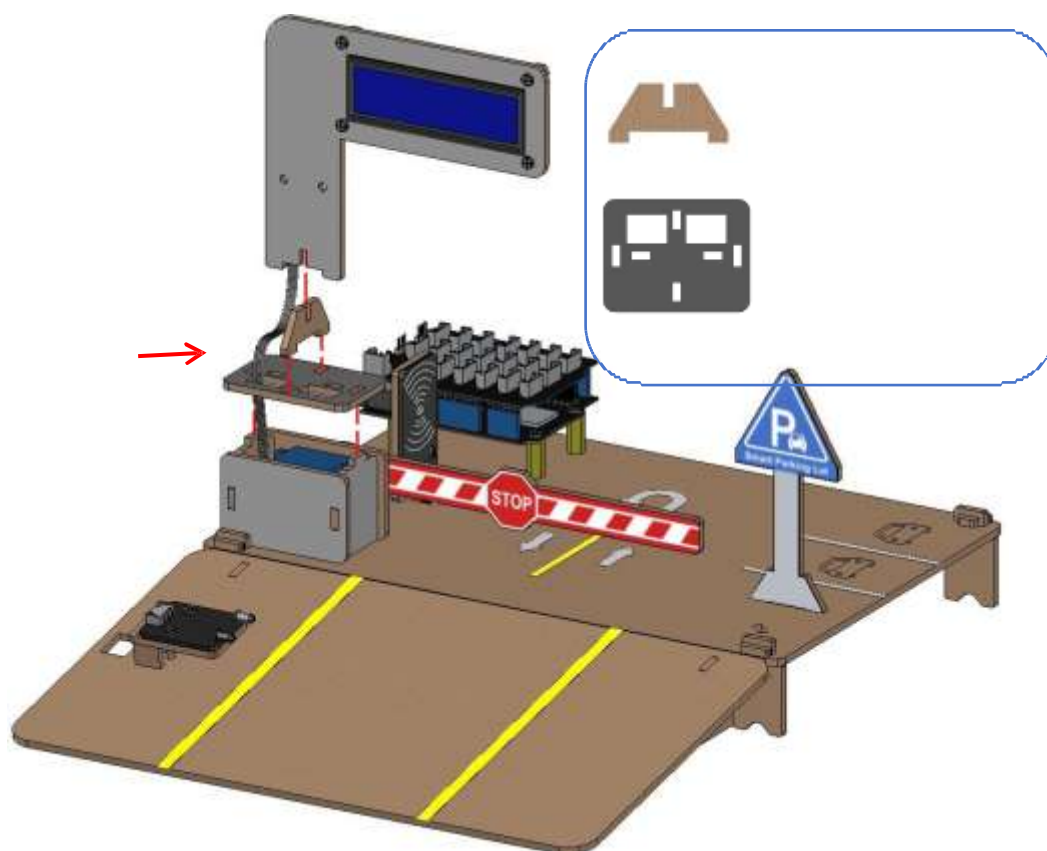


M3*12MM Round Head Screws x 4



Bracket board x 1

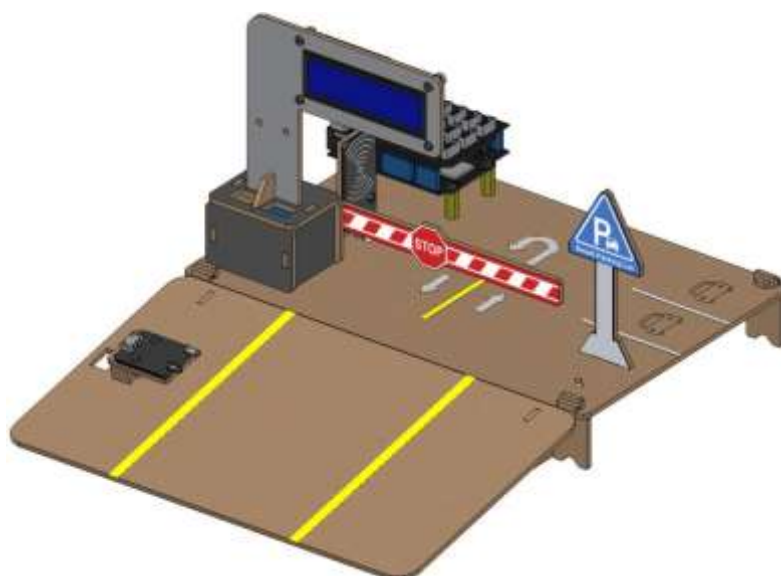
Krok 2: Upevněte LCD displej na základnu.



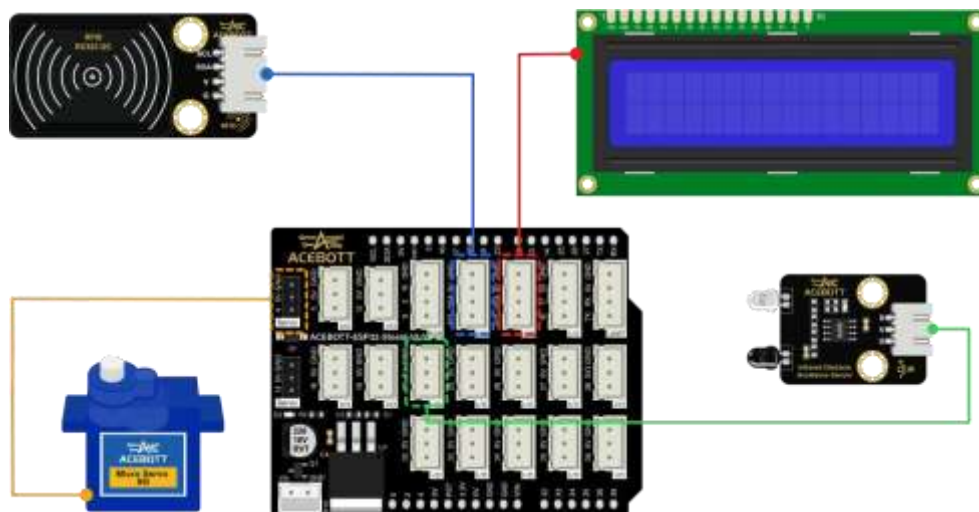
Bracket board x 1

Servo box No.5 board x 1

Protáhněte Dupontovu vodičku štěrbinami po obou stranách serva.



Krok 3: Zapojení.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
RFID module	SCL	SCL
	SDA	SDA
	VCC	5V
	GND	GND
Servo	S	4
	V	5V
	G	GND
Infrared obstacle avoidance sensor	S	23
	V	5V
	G	GND

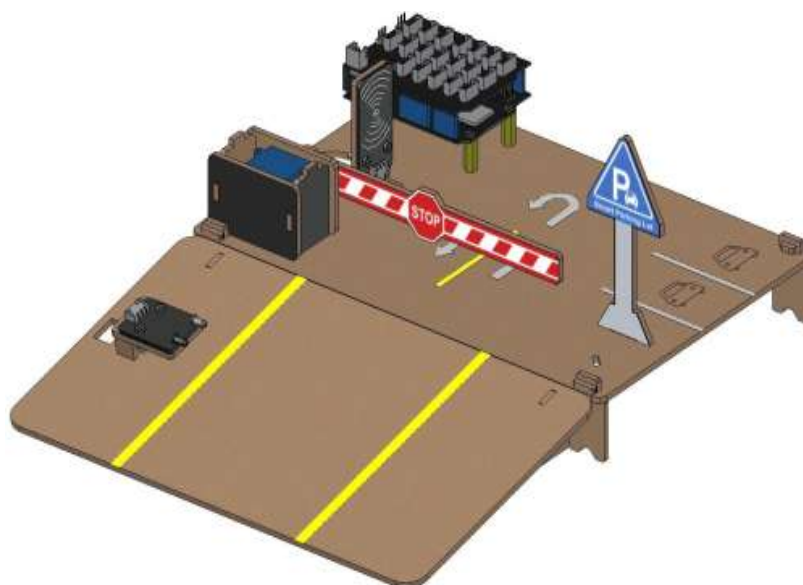
LCD1602 display screen	SCL	SCL
	SDA	SDA
	VCC	5V
	GND	GND

Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

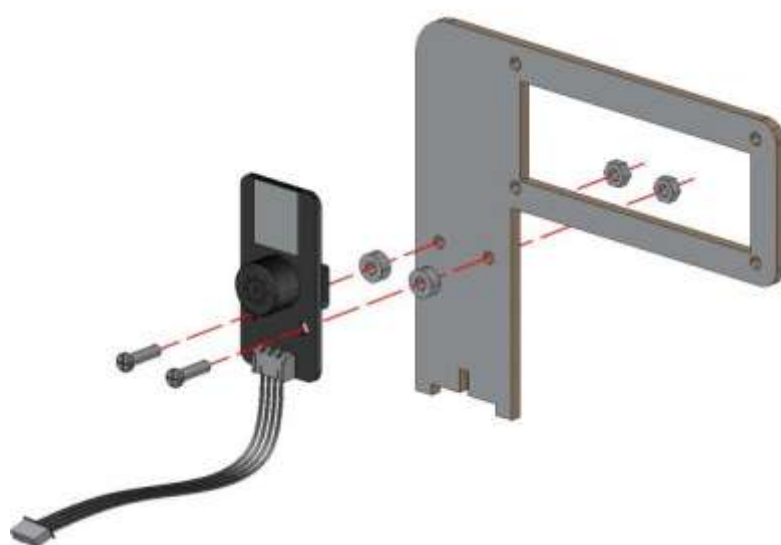
13 Hlasový systém vysílání

Příjemné upozornění: Tento projekt bude pokračovat ve struktuře „inteligentního řízení vstupu a výstupu“ (jak je znázorněno níže), zde nebudou znázorněny konkrétní kroky stavby, ale přejdeme přímo k sestavení modulu Mp3.



Poznámka: Před instalací modulu MP3 postupujte podle pokynů v učebnici a nainportujte MP3 soubory, které mají být přehrávány, do paměťové karty modulu MP3. [Klikněte zde pro stažení MP3 hudebních souborů.](#)

Krok 1: Nainstalujte modul pro přehrávání MP3.



**MP3 playback
module x 1**

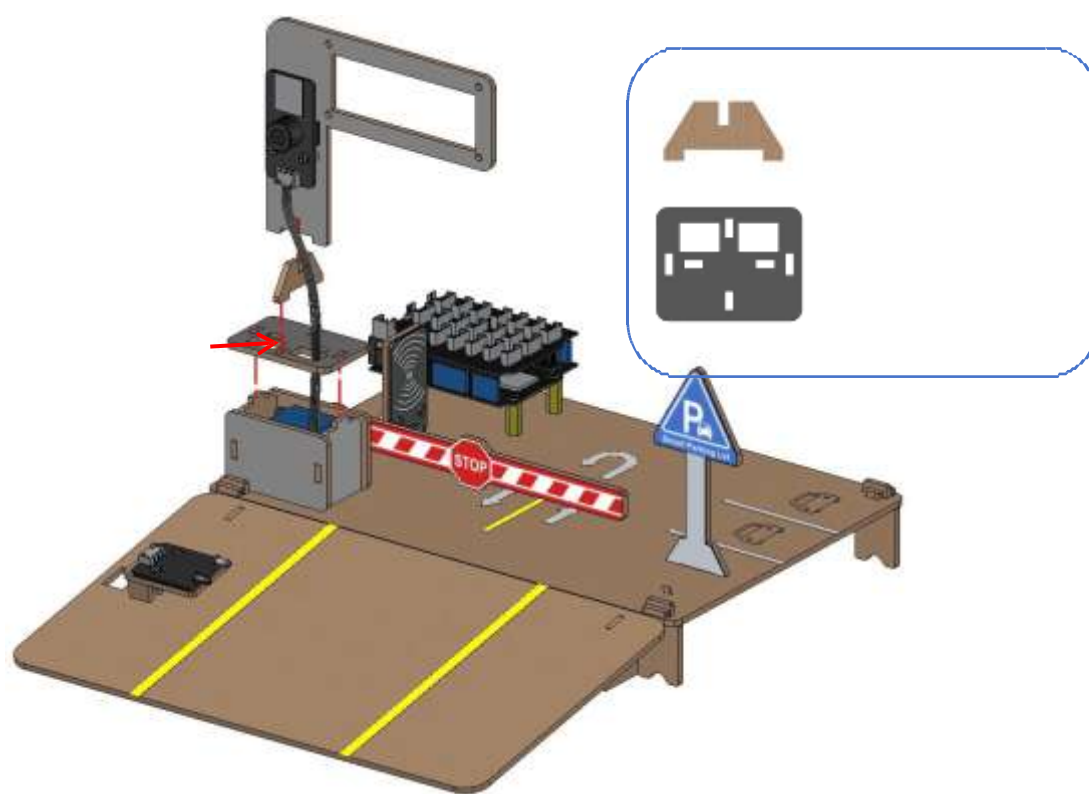
M3*3 gasket x 2

**M3 Nickel-Plated
Nuts x 2**

**M3*12MM Round
Head Screws x 2**

Bracket board x 1

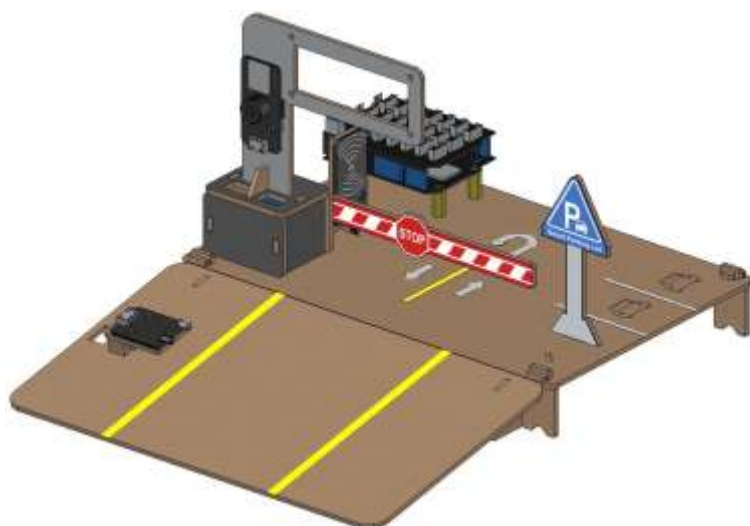
Krok 2: Připevněte modul pro přehrávání MP3 na základnu.



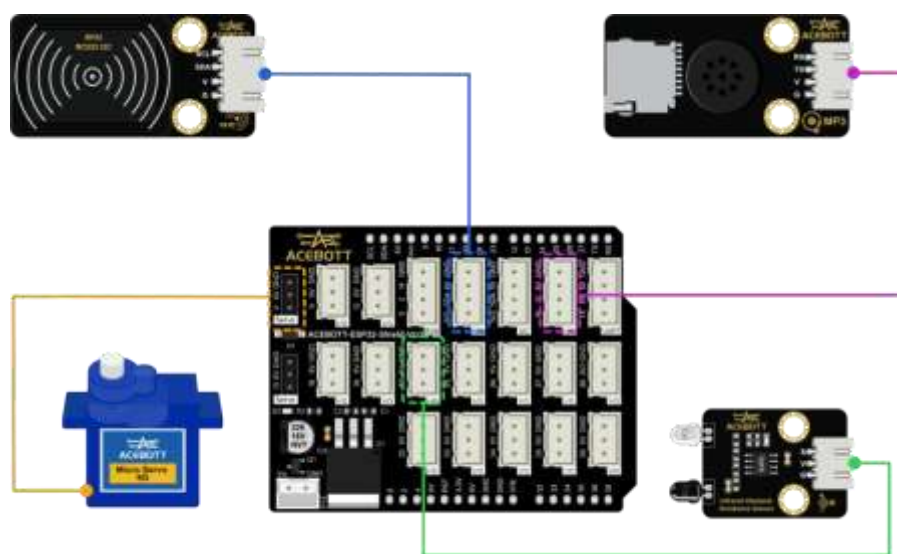
Bracket board x 1

Servo box No.5 board x 1

Protáhněte Dupontovu vodičku štěrbinami po obou stranách serva.



Krok 3: Zapojení.



Pokyny pro zapojení:

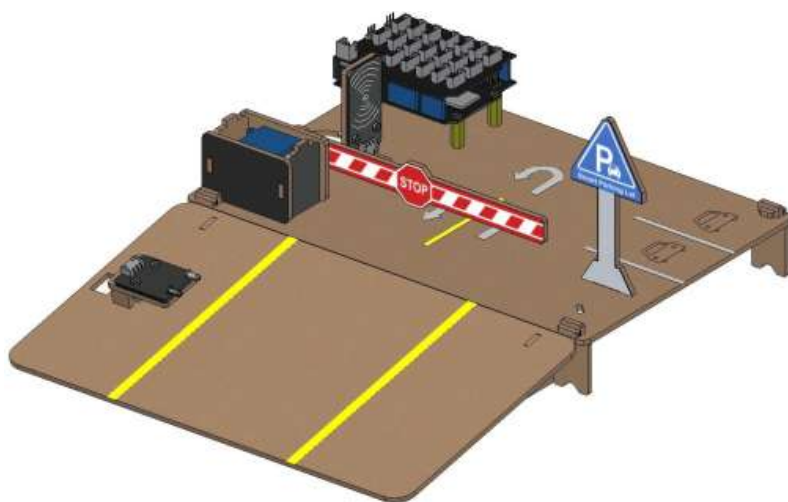
Název modulu	Pin modulu	Pin ESP32
RFID module	SCL	SCL
	SDA	SDA
	VCC	5V
	GND	GND
Servo motor	S	4
	V	5V
	G	GND
MP3 playback module	TX	16
	RX	17
	V	5V
	G	GND
Infrared obstacle avoidance sensor	S	23
	V	5V
	G	GND

Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

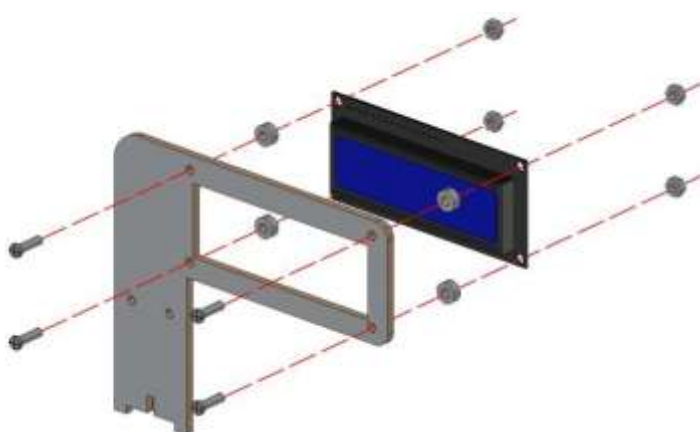
[Klikněte zde pro otevření programu pro tento projekt.](#)

14 Inteligentní parkovací systém integrován

Upozornění: Tento projekt bude pokračovat ve struktuře „inteligentního řízení vstupu a výstupu“ (jak je znázorněno na následujícím obrázku), zde již nebudou ukazovány konkrétní kroky montáže, ale přímo se začne s montáží LCD displeje a MP3 modulu.



Krok 1: Nainstalujte LCD displej.



LCD1602 module x 1



M3*3 gasket x 4



M3 Nickel-Plated Nuts x 4



M3*12MM Round Head Screws x 4



Bracket board x 1

Krok 2: Nainstalujte modul MP3 přehrávače.



MP3 playback module x 1



M3*3 gasket x 2



M3 Nickel-Plated Nuts x 2



M3*12MM Round Head Screws x 2



4P connection line x 1

Krok 3: Nainstalujte horní desku serva.

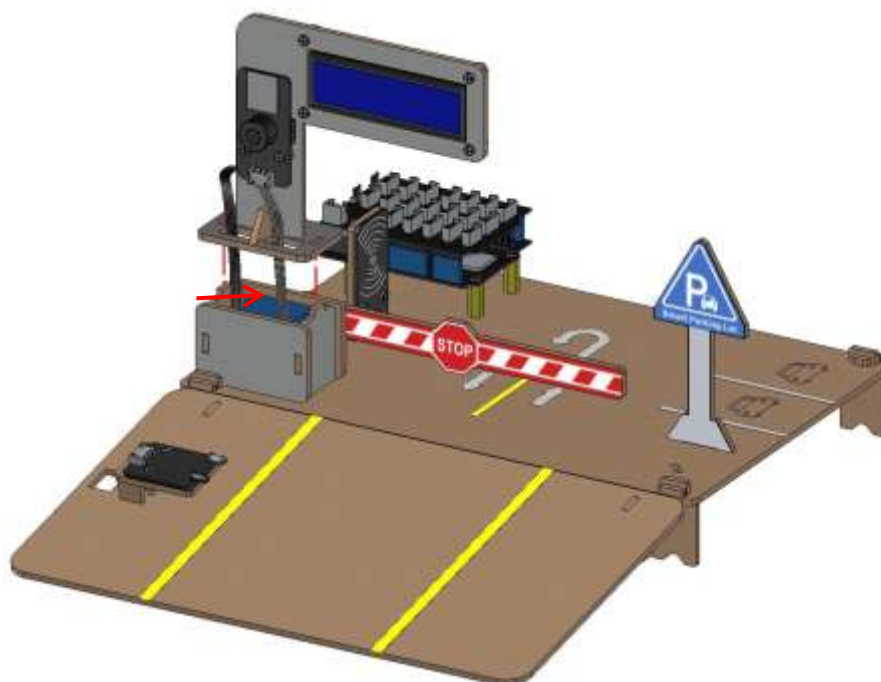


Bracket board x 1



Servo box No.5 board x 1

Krok 4: Upevněte horní desku serva.



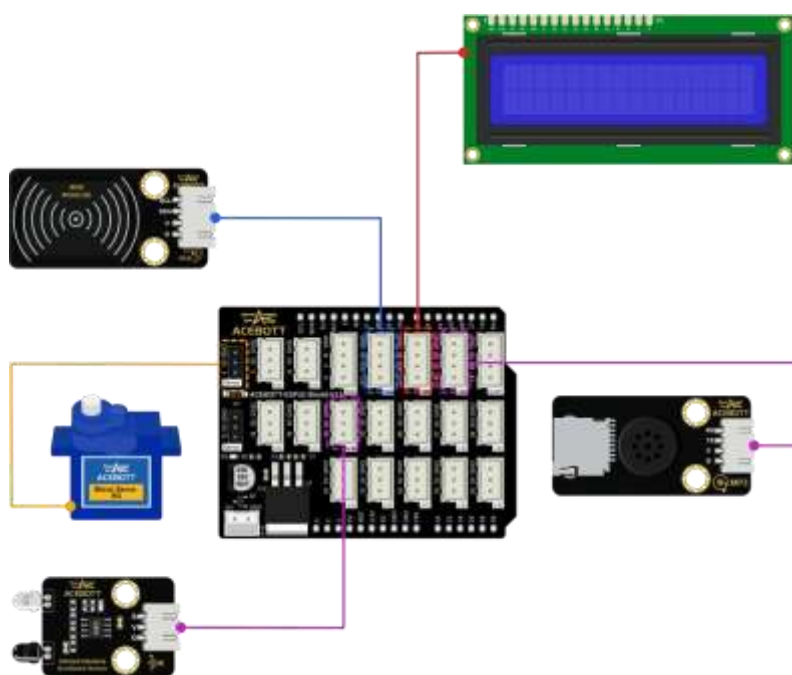
Protáhněte Dupontovu vodičku štěrbinami po obou stranách serva.

Krok 5: Nainstalujte sochu LUMIHO.





Krok 6: Kabeláž.



Pokyny pro zapojení:

Název modulu	Pin modulu	Pin ESP32
RFID module	SCL	SCL
	SDA	SDA
	VCC	5V
	GND	GND
Servo motor	S	4
	V	5V
	G	GND
MP3 playback module	TX	16
	RX	17
	V	5V
	G	GND
Infrared obstacle avoidance sensor	S	23
	V	5V
	G	GND
LCD1602 display screen	SCL	SCL
	SDA	SDA
	VCC	5V
	GND	GND

Poznámka: Při připojování modulu k ovládací desce ESP32 důsledně dodržujte pokyny k zapojení. Nesprávné zapojení může způsobit zkrat a poškodit ovládací desku ESP32.

[Klikněte zde pro otevření programu pro tento projekt.](#)

Kapitola 1 Inteligentní doprava

V této kapitole se budeme zabývat vývojem a aplikacemi inteligentních dopravních technologií a diskutovat o jejich dopadu na lidstvo. Dále se podíváme na aplikaci mikrokontrolérů v inteligentních dopravních systémech, konkrétně se zaměříme na mikrokontrolér ESP32, abychom se důkladně připravili na nadcházející obsah.



Sekce 1: Přehled inteligentní dopravy



Cíle



-  **Pochopit koncept inteligentní dopravy**
-  **Pochopit vývoj inteligentní dopravy**
-  **Pochopit aplikace inteligentní dopravy**
-  **Prozkoumat vliv inteligentní dopravy na lidský život**



Doprava je součástí našeho každodenního života. Narazili jste při cestování na nějaké problémy? Jak bychom měli tyto problémy řešit?

S rychlým rozvojem ekonomiky a technologie, urbanizace roste a počet vozidel na silnicích se zvyšuje. To vedlo k mnoha běžným dopravním problémům, jako jsou zácpy, časté nehody, znečištění životního prostředí a nedostatek energie. Aby se těmto problémům čelilo a zlepšila



situace, lidé vylepšili dopravní systém různými technologickými prostředky, což vedlo k vzniku inteligentních dopravních systémů.

Inteligentní doprava



Inteligentní dopravní systémy (ITS) se týkají optimalizace a inteligentní transformace dopravních systémů pomocí pokročilých informačních technologií, komunikačních technologií a umělé inteligence. Díky technologii inteligentní dopravy může dopravní systém v reálném čase získávat, analyzovat a využívat různé dopravní údaje.

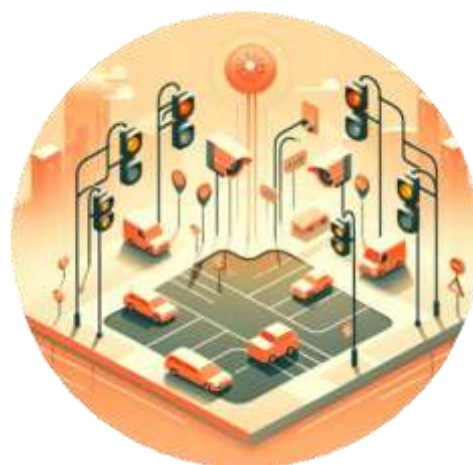
To optimalizuje dopravní tok, zlepšuje efektivitu dopravy, zvyšuje bezpečnost a snižuje spotřebu energie, čímž poskytuje obyvatelům měst pohodlnější, bezpečnější a ekologičtější možnosti cestování. Inteligentní doprava zahrnuje širokou škálu technologických aplikací, včetně inteligentního řízení dopravních signálů, inteligentních systémů řízení dopravy, inteligentních systémů monitorování a varování v dopravě a inteligentních dopravních informačních služeb. Konečným cílem je dosáhnout automatizace a inteligence v dopravních systémech a poskytovat vědecká, přesná a efektivní řešení pro městskou správu dopravy.

Vývoj inteligentní dopravy



Raný vývoj inteligentních dopravních systémů se primárně spoléhal na základní elektronické a počítačové technologie, přičemž se zaměřil na zlepšení řízení dopravních signálů a správu toku vozidel. Například systémy řízení dopravních signálů byly používány k optimalizaci kapacity

křižovatek nebo byla využívána uzavřená televizní okružová (CCTV) k monitorování dopravních podmínek na hlavních silnicích.



Jak se vyvinula senzorová technologie, bezdrátová komunikace a algoritmy umělé inteligence, začaly se inteligentní dopravní systémy integrovat o pokročilejší funkce, jako jsou automatické identifikační systémy (například elektronický výběr mýtného - ETC), služby v reálném čase pro poskytování dopravních informací a systémy pro

prioritu nouzových vozidel. Aplikace těchto technologií významně zlepšila efektivitu dopravních systémů, snížila úroveň zácp a nehod a zlepšila cestovní zkušenosti uživatelů.

V budoucnu, s technologickým pokrokem a zrychlenou urbanizací, můžeme očekávat inteligentnější, efektivnější, bezpečnější a ekologičtější dopravní ekosystém. Budoucí inteligentní doprava bude hluboce integrovat Internet věcí, umělou inteligenci, analýzu velkých dat a technologie autonomního řízení k



dosažení komplexní vzájemné propojenosti mezi vozidly, infrastrukturou a lidmi. To nejen významně zlepší operační efektivitu dopravních systémů, sníží zácpy a nehody, ale také poskytne uživatelům personalizovanější a pohodlnější cestovní zážitek. Kromě toho, jak pokračuje tlak na cíle udržitelného rozvoje, se budoucí inteligentní doprava zaměří také na rozvoj zelených dopravních prostředků a energetických zdrojů, jako jsou elektrická vozidla a vozidla na vodíkové palivové články, stejně jako na snižování spotřeby energie a emisí CO₂ pomocí inteligentního plánování tras a harmonogramů, čímž přispěje k budování udržitelného městského dopravního prostředí.



Inteligentní dopravní zařízení: Pomocí pokročilých technologií jsou dopravní zařízení řízena inteligentně. Například inteligentní pouliční osvětlení může automaticky upravit svou jasnost na základě dopravního toku a změn světla, což efektivně šetří

energii a zvyšuje bezpečnost noční jízdy; systémy dopravních signálů mohou upravit dobu trvání signálů na základě aktuálních dopravních podmínek, aby optimalizovaly tok vozidel a snížily zácpy; semaforey pro chodce mohou automaticky upravit dopravní signály podle počtu chodců, aby zajistily bezpečný přechod.

Inteligentní řízení dopravy: Řízení dopravy je dosaženo prostřednictvím senzorů a automatizačních technologií. Například systémy pro detekci dopravního toku používají senzory a kamery k monitorování aktuálního objemu dopravy a zácp na

silnicích, což pomáhá dopravním správám při předpovědi dopravního toku a optimalizačním řízení; inteligentní mýtné stanice využívají automatickou identifikaci a bezkontaktní platební technologie, aby zlepšily efektivitu výběru



mýtného, zkrátily dobu čekání vozidel ve frontách a optimalizovaly provozní efektivitu dopravy.



Inteligentní technologie vozidel: Funkce vozidel jsou vylepšeny použitím senzorů a technologií umělé inteligence. Například technologie autonomního řízení umožňuje vozidlům samostatně rozpoznávat dopravní signály a překážky, čímž dosahují automatické navigace a jízdy, což zvyšuje bezpečnost jízdy a

efektivitu dopravy. Technologie inteligentní komunikace vozidel také umožňuje vozidlům propojit se s dopravní infrastrukturou a dalšími vozidly, aby sdílely aktuální dopravní podmínky, což pomáhá řidičům vyhnout se zácpám a nehodám.

Aplikace těchto inteligentních dopravních technologií nejen zvýšila úroveň inteligence v řízení městské dopravy, ale také výrazně zlepšila cestovní zážitky obyvatel měst, poskytujícím důležitou podporu pro bezpečnost, efektivitu a pohodlí městské dopravy.



Jaká inteligentní dopravní zařízení jsou ve vašem městě? Jaké pohodlí přinášejí?

Ačkoli inteligentní dopravní systémy dosáhly významného pokroku při podpoře rozvoje měst, stále čelí několika výzvám a problémům. Za prvé, chybí standardizace a vzájemná kompatibilita mezi inteligentními dopravními systémy v různých regionech a městech, což vede k překážkám v sdílení informací a spolupráci. Za druhé, výstavba a provoz inteligentních dopravních systémů vyžadují značnou finanční a technickou podporu, a mnoho oblastí postrádá dostatečné zdroje a schopnosti pro jejich propagaci a aplikaci. Kromě toho je třeba posílit bezpečnostní a ochranu soukromí inteligentních dopravních systémů. I přes tyto výzvy má inteligentní doprava stále velký potenciál pro rozvoj a pokrok, což slibuje přinést více inovací a vitality do budoucnosti měst.



Pojďme přemýšlet o tom, jak vyřešit současné problémy s inteligentními dopravními systémy. Zároveň si představme, jakými směry se může inteligentní doprava dále rozvíjet. Napište své nápady do rámečku níže.

Jak vyřešit současné problémy
inteligentní dopravy:

Směr budoucího vývoje
inteligentní dopravy:

Sekce 2: Mozek inteligentní dopravy



Cíle kurzu



-  **Pochopit koncept mikrokontrolérů**
-  **Porozumět struktuře a funkci ESP32**
-  **Pochopení funkce Arduino IDE**
-  **Osvojení používání Arduino IDE**
-  **Zkušenosti s psáním a stahováním programu pro ESP32**



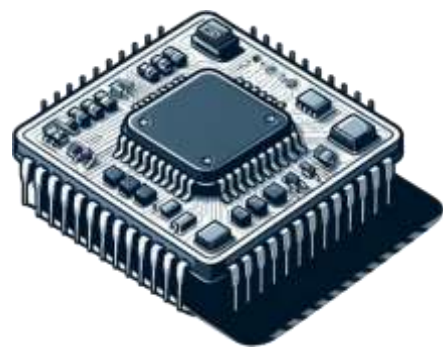
Když dopravní systém čelí tisícům dopravních dat, jak je efektivně zpracovává?

Důvodem, proč jsou inteligentní dopravní systémy inteligentní, je to, že mají „mozek“, kterým je ovladač. Tento ovladač dokáže rychle zpracovávat informace, rozumně přidělovat zdroje a zajišťovat plynulý provoz celého dopravního systému. Tento „mozek“ může mít různé podoby, jako je počítač, mikročip nebo zařízení typu gateway. Jeho hlavní funkcí je přijímat informace prostřednictvím programování, analyzovat a zpracovávat je a následně vydávat příslušné ovládací příkazy, čímž se dosahuje automatizace a inteligence řízení dopravy. V nadcházejících kurzech použijeme mikročip jako tento „mozek“, který nám pomůže lépe porozumět a aplikovat inteligentní dopravní

Jednočip



Jednočipový mikropočítač (SCM) je integrovaný čip, který kombinuje jádro mikroprocesoru s různými funkcemi, jako je paměť (například RAM a ROM), vstupní/výstupní porty (I/O), časovače atd., čímž vytváří kompletní mikropočítačový systém.



SCM se vyznačují kompaktní velikostí, nákladovou efektivitou, snadnou programovatelností a integrací, což je činí široce využívanými v různých oblastech života, například v domácích spotřebičích, průmyslovém řízení, automobilové elektronice, lékařském vybavení atd.

Existuje mnoho typů SCM, a proto můžeme vybrat vhodný model podle našich potřeb. Model SCM používaný v tomto kurzu je ESP32.

ESP32



ESP32 je nízkonapěťový, vysoce integrovaný mikrořadič s dvojitým režimem Wi-Fi a Bluetooth, který byl vyvinut společností Espressif Systems. Vývojová deska ESP32 je velmi populární díky svým výkonným funkcím a hojným komunikačním rozhraním, což ji činí vhodnou pro různé aplikace Internetu věcí (IoT), vývoj vestavěných systémů a implementace projektů. Níže je uveden schématický diagram řídicí desky ACEBOTT-ESP32:



Vstup napájení:

Může být připojen k napájecímu zdroji o napětí 7,2V-15V pro napájení řídicí desky.



Typ-C rozhraní:

Pro stažení programu může také poskytovat 5V napájení pro řídicí desku.



Tlačítko pro resetování:

Resetuje systém řídicí desky.



GPIO: Obecné vstupní a výstupní rozhraní pro vstup a výstup dat.



Moduly ESP32:

Obsahuje mikroprocesor ESP32, Bluetooth, WiFi.

Po získání řídicí desky ESP32 je její počáteční stav nevyužitelný pro konkrétní funkce. Pokud chcete, aby měla konkrétní funkci, musíte ji

naprogramovat. To vyžaduje nejprve konfiguraci vývojového prostředí vhodného pro ESP32, které poskytne platformu pro programování pro vývojáře. Mezi běžně používané vývojové platformy pro ESP32 patří MicroPython, Arduino IDE, ESP-IDF, ACECode a další. Tento kurz se zaměří na použití ACECode jako vývojové platformy pro ESP32.

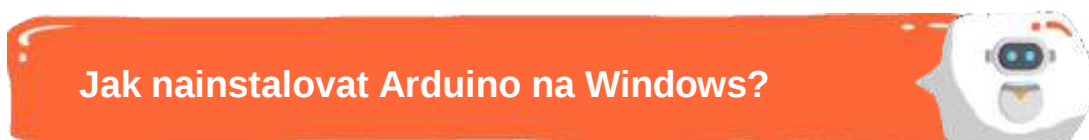


Arduino IDE je bezplatný multiplatformní nástroj pro programování, který slouží k psaní, nahrávání a ladění kódu pro Arduino. Kód pro Arduino je napsán na základě programovacího jazyka C++, který nabízí mnoho pokročilých knihoven a funkcí, což zjednodušuje ovládání a interakci s hardwarem, což usnadňuje začátečnickům učení.



Arduino IDE může běžet na operačních systémech Windows, Mac a Linux. V tomto případě se zaměříme na to, jak nainstalovat Arduino IDE na Windows.

Arduino IDE



Jako software s otevřeným zdrojovým kódem je Arduino IDE vyvinuto na základě Processing IDE, což je integrované vývojové prostředí oficiálně

spuštěné společností Arduino.

S Arduino IDE stačí napsat kód programu do IDE a poté jej nahrát na desku Arduino a program řekne desce Arduino, co má udělat.

Stáhněte si Arduino IDE pro Windows

1) Stáhněte si Arduino IDE

① Stáhněte si webovou adresu Arduino IDE: <https://www.arduino.cc/en/Main/Software>, otevřete adresu URL podle následující tabulky a v budoucnu vyberte odpovídající verzi. Verze softwaru může být aktualizována, stačí nainstalovat nejnovější verzi.

Arduino IDE 2.2.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

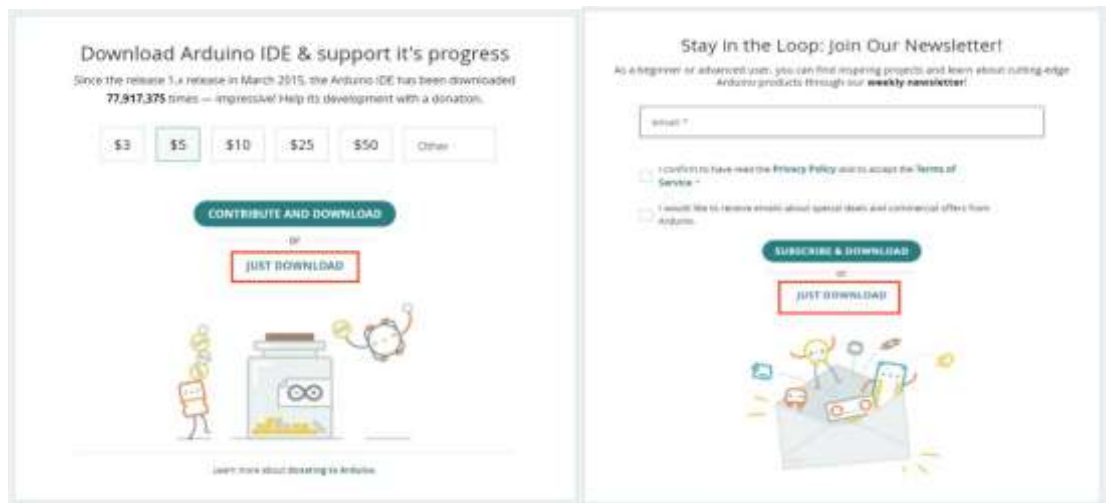
The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (x86-64)
- Linux** ZIP file 64 bits (x86-64)
- macOS** Intel, 10.14: "Mojave" or newer, 64 bits
- macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

② Vyberte JUST DOWNLOAD.

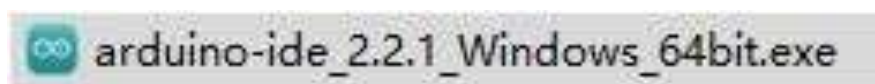


③ Když se objeví následující obrazovka, Arduino IDE se stahuje.



Nainstalujte Arduino IDE

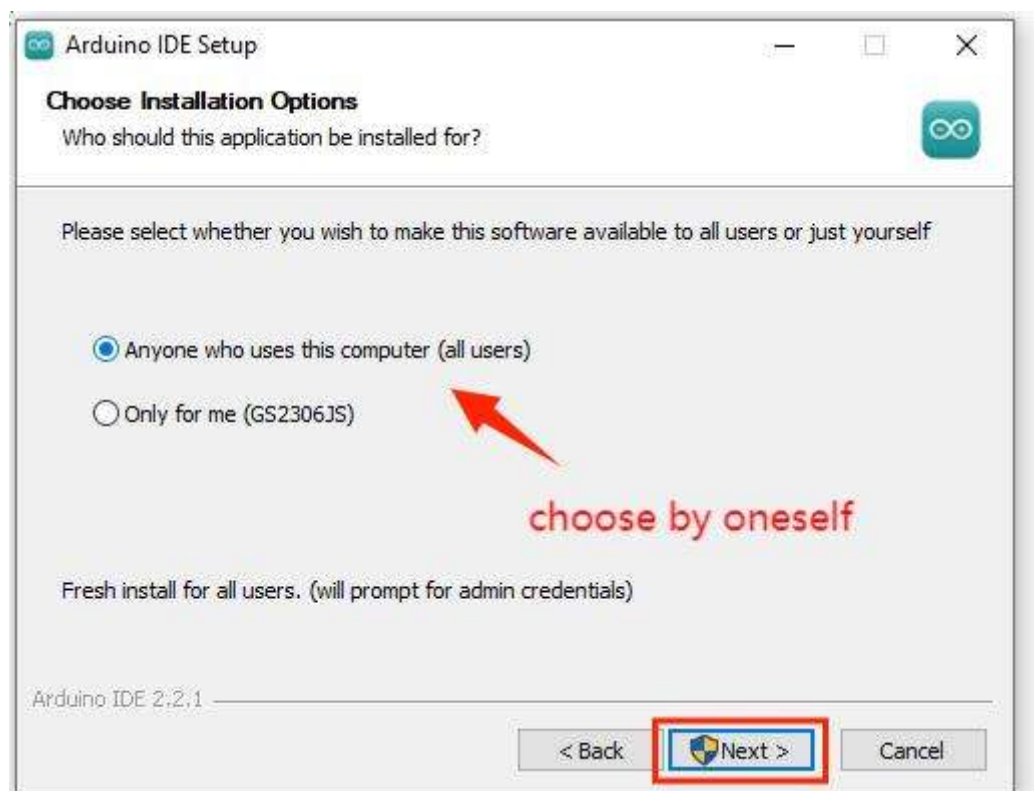
① Po dokončení stahování se zobrazí soubor ikony. Klepněte na Instalovat software.



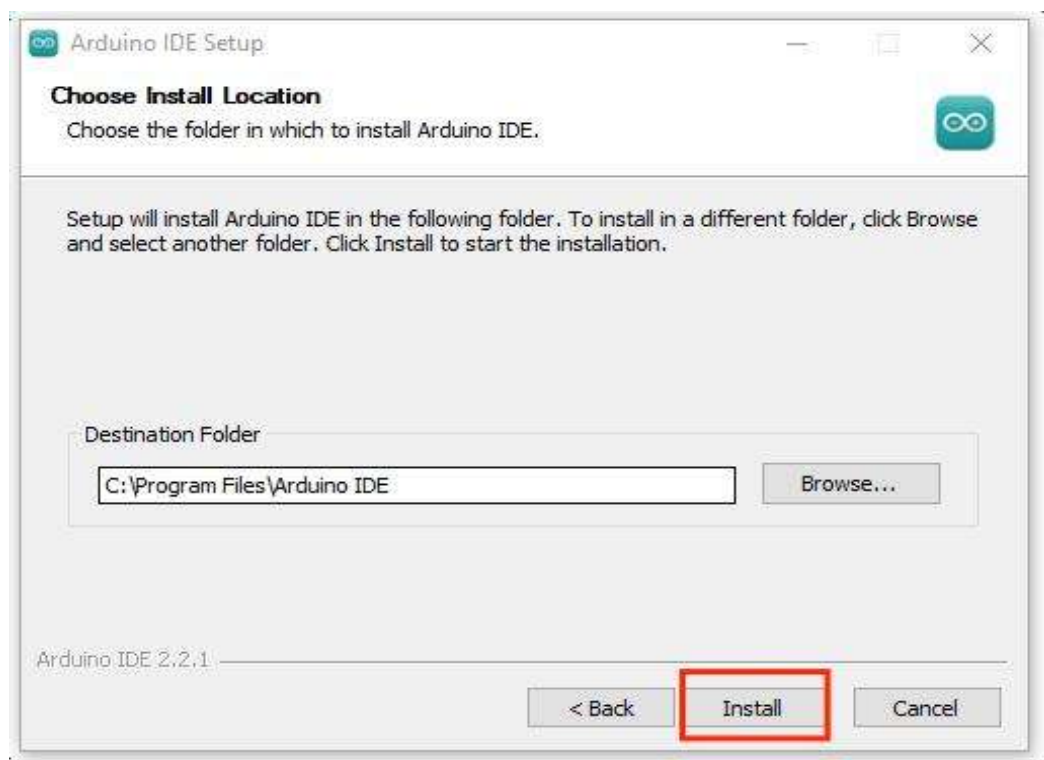
② Po instalaci se zobrazí následující obrazovka a vyberte „I Agree“.



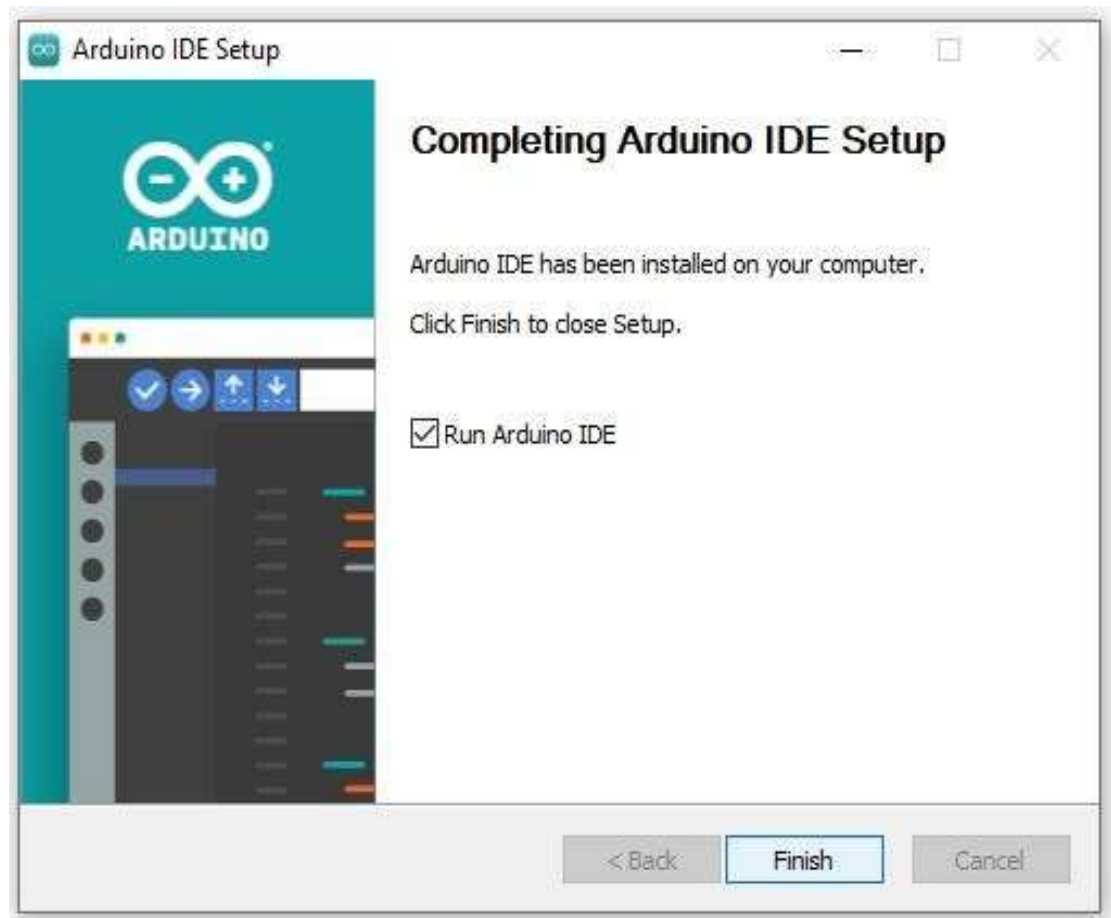
③ Po výběru „Next“ se zobrazí následující obrazovka a vyberte „Install“.



- ④ Vyberte „Next“ a objeví se následující obrazovka. Vyberte "Install".



⑤ Nainstalovaný software Arduino IDE.



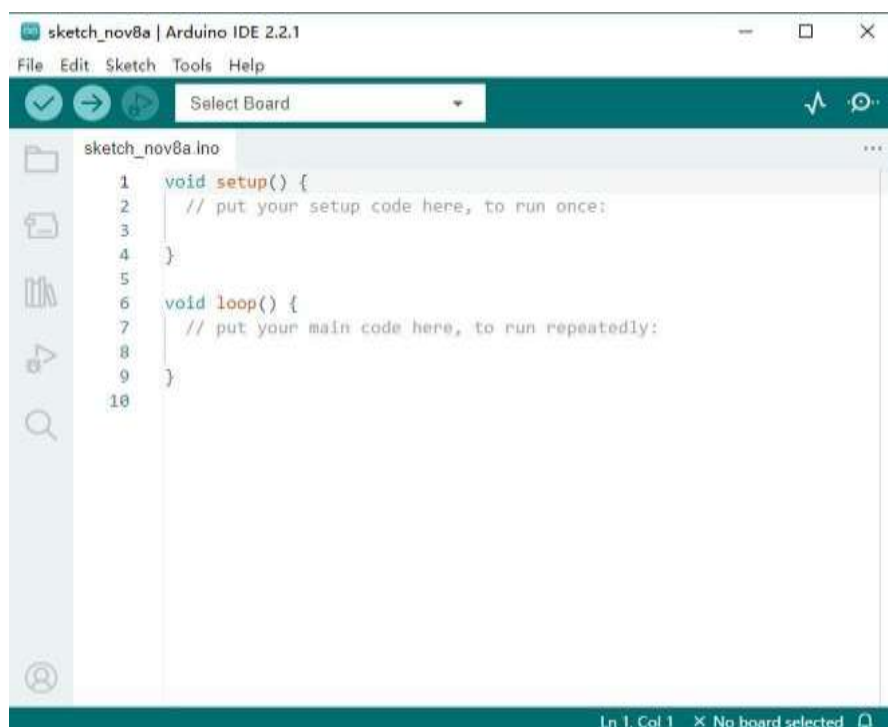
⑥ Instalace je dokončena.

2) Instalace Arduino IDE je dokončena

① Po instalaci se na ploše objeví ikona zástupce Arduino IDE.



② Po otevření se zobrazí následující obrazovka.



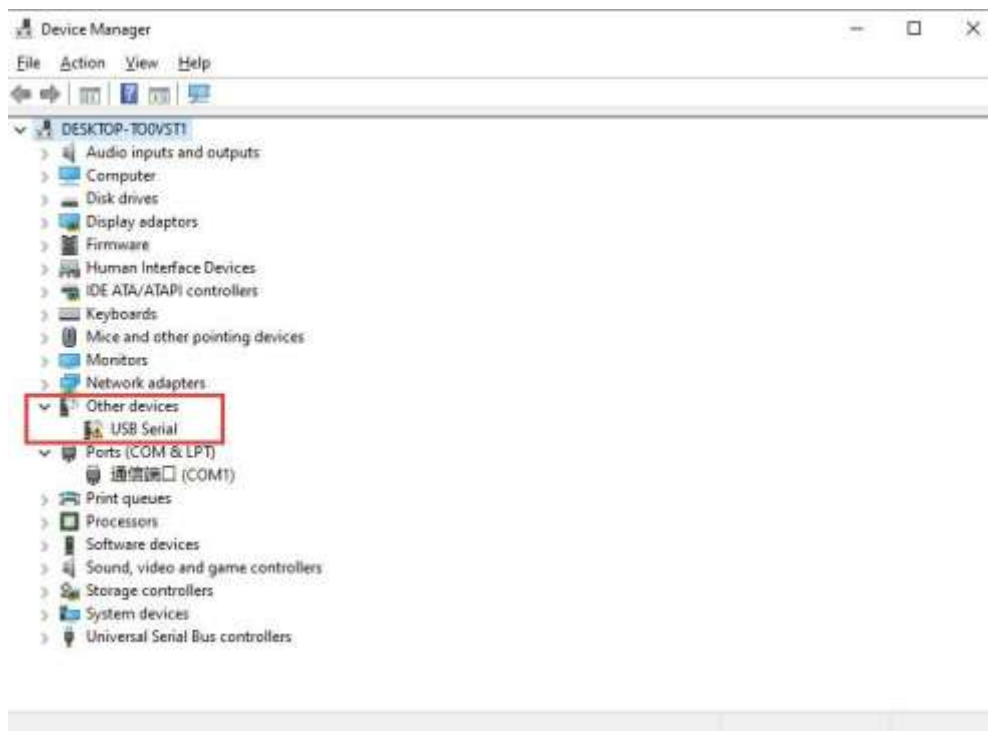
Jak nainstalovat ovladač sériového portu CH340

Připojte hlavní řídicí desku k počítači pomocí kabelu USB a ovladač se automaticky nainstaluje na systémy MacOS a Windows. Pokud se instalace ovladače nezdaří, musíte ovladač nainstalovat ručně.

Čip USB do sériového portu řídicí desky ESP32 je CH340C. Proto je potřeba nainstalovat ovladač pro čip. Proces instalace ovladače je na různých systémech v podstatě stejný. Zde si ukážeme instalaci ovladače na systému Win10. Složku "[USB Drive CH341 3 1](#)" naleznete v balíčku zdrojů, který jsme vám poskytli. Toto je soubor ovladače, který chceme nainstalovat.

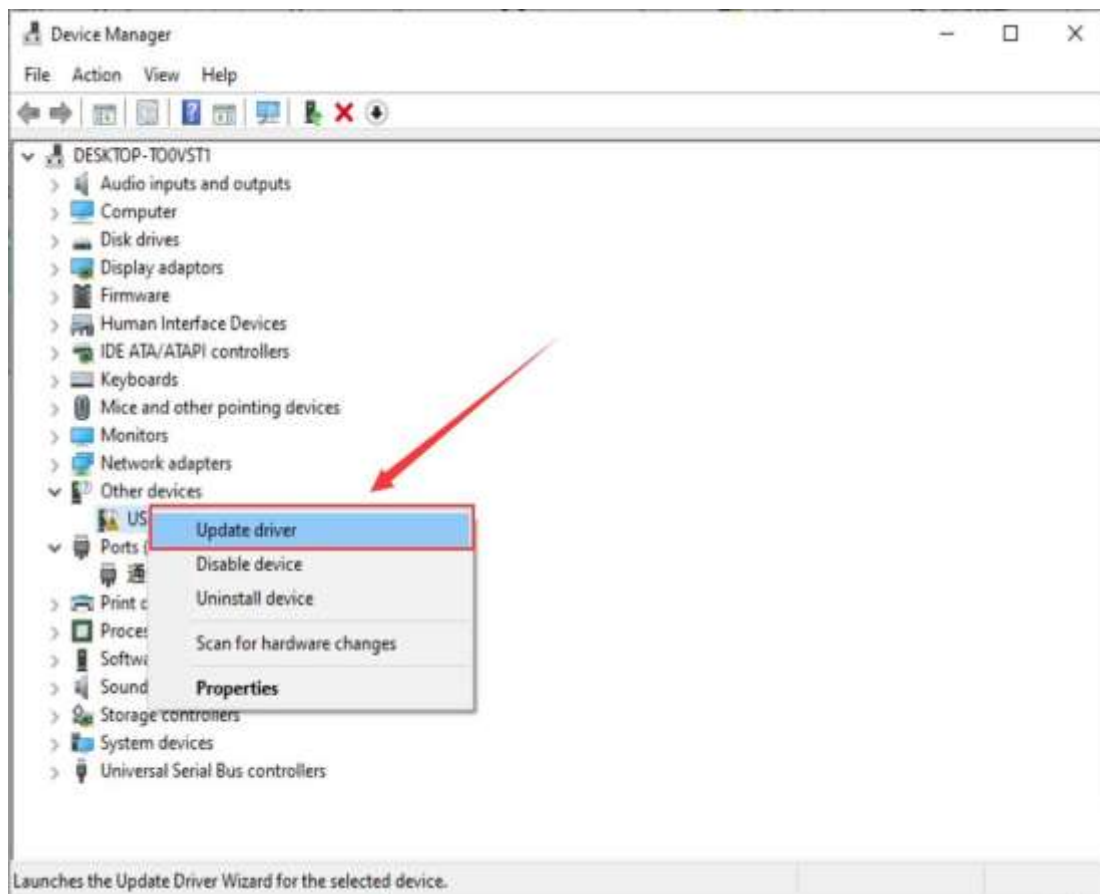
1) Zkontrolujte, zda je nainstalován ovladač sériového portu CH340 (pokud je nainstalován, přeskočte jej)

- ① Zapojte jeden konec USB kabelu do řídicí desky ESP32 a druhý konec do USB portu na počítači.
- ② Při prvním připojení konzole ESP32 k počítači klikněte pravým tlačítkem na "My Computer" -> "Attribute" -> Klikněte na "Device Manager" a pod "Other Devices" uvidíte buď "USB-Serial" nebo "Unknown Device".

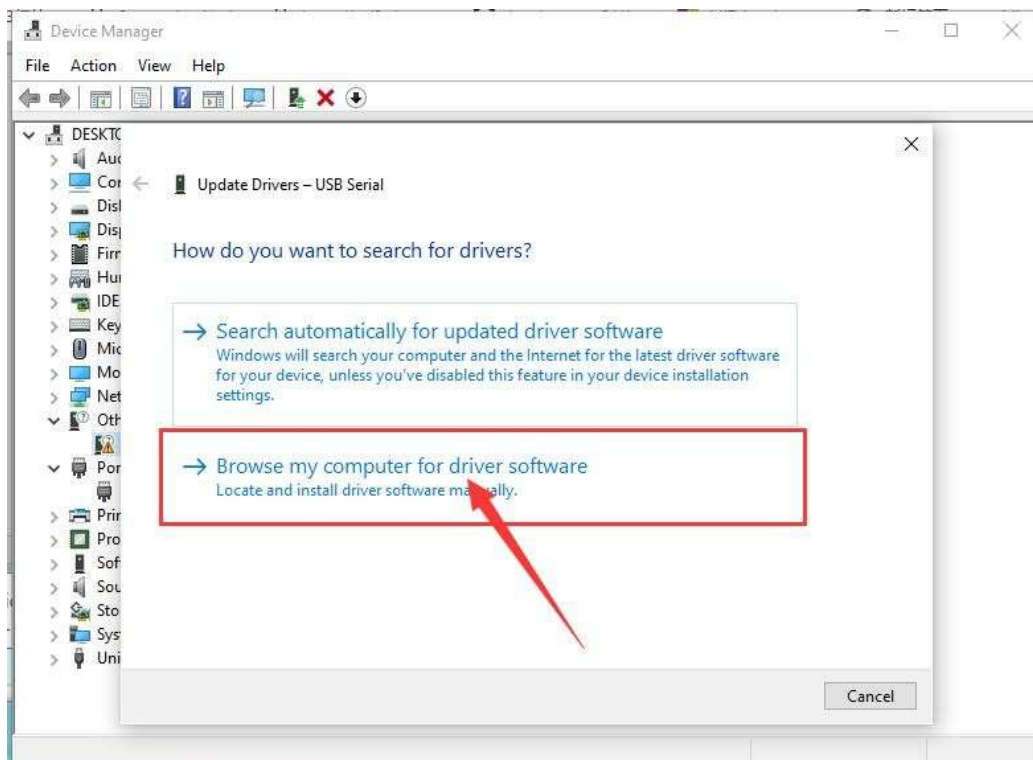


2) Nainstalujte ovladač sériového portu CH340

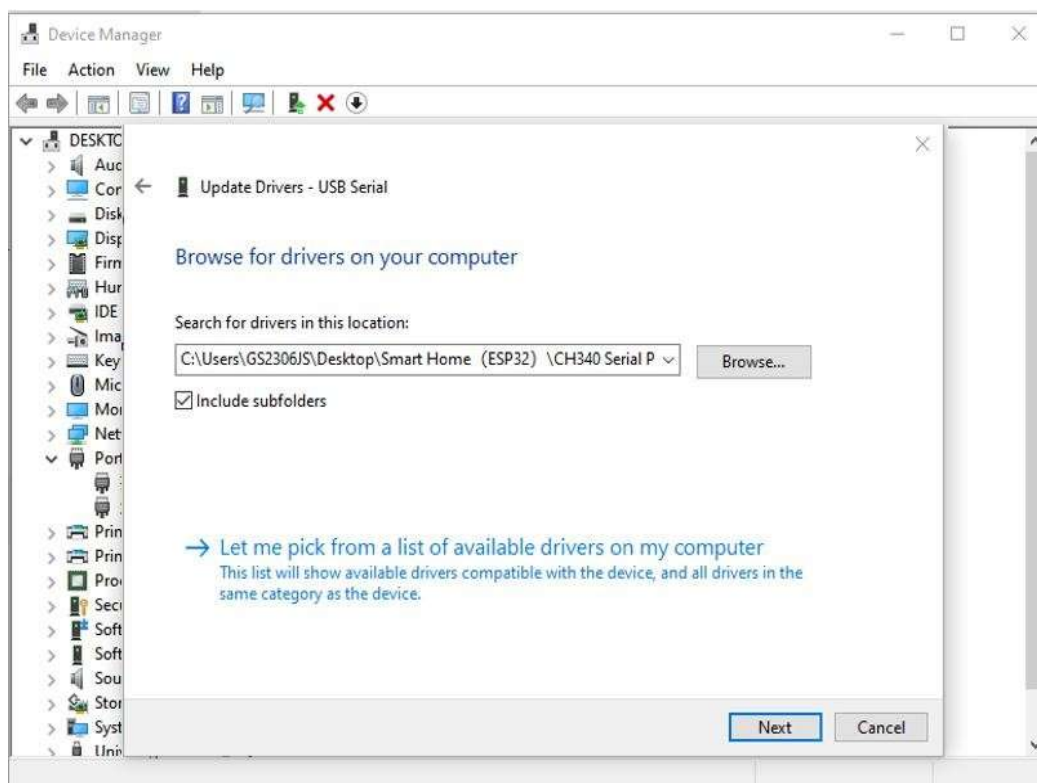
- ① Nejprve klikněte pravým tlačítkem na zařízení a vyberte možnost horní nabídky (Aktualizovat software ovladače), jak je uvedeno níže.



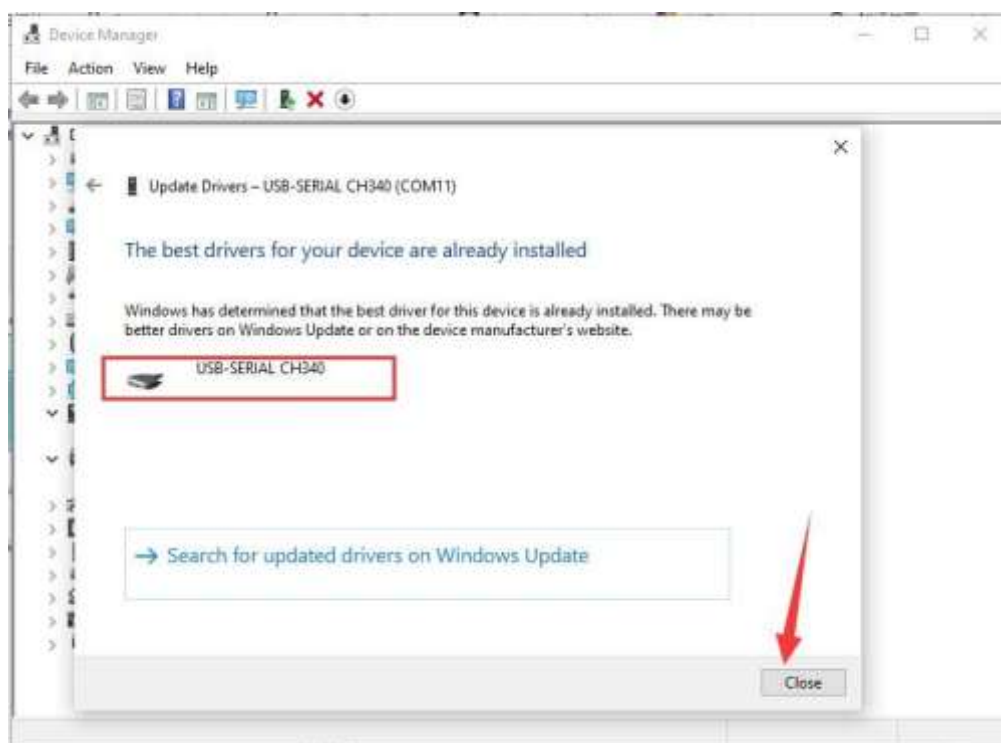
② Poté budete vyzváni k „Automatically search for updated driver software“ nebo „Browse my computer for driver software“, jak je uvedeno níže. Na této stránce vyberte „Browse my computer for driver software“.



③ Poté přidejte cestu k souboru ovladače.

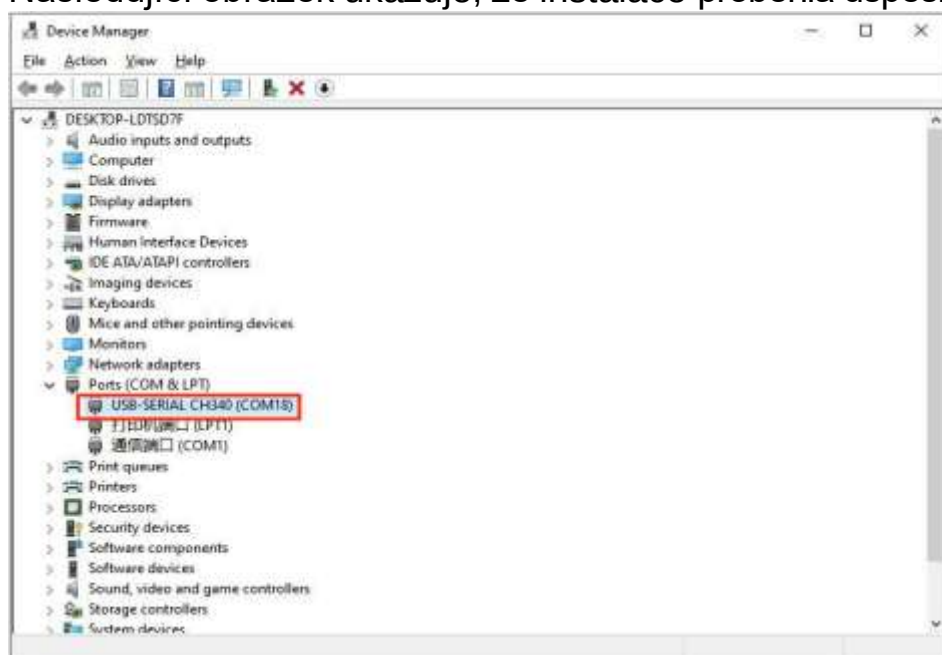


④ Po dokončení instalace softwaru obdržíte potvrzovací zprávu. Po dokončení instalace klikněte na „Close“.



3) Potvrďte, že byl ovladač sériového portu CH340 úspěšně nainstalován

Zapojte jeden konec USB kabelu do řídicí desky ESP32 a druhý konec do USB portu na počítači. Klikněte pravým tlačítkem na "My Computer" -> "Attribute" -> Klikněte na "Device Manager" a připojte se k ovládacímu panelu. Následující obrázek ukazuje, že instalace proběhla úspěšně.



Výukový program pro Mac OS pro instalaci Arduino IDE

Arduino IDE

Jako software s otevřeným zdrojovým kódem je Arduino IDE vyvinuto na základě Processing IDE, což je integrované vývojové prostředí oficiálně spuštěné společností Arduino.

S Arduino IDE stačí napsat kód programu do IDE a poté jej nahrát na desku Arduino. Program řekne Arduino, co má deska dělat.

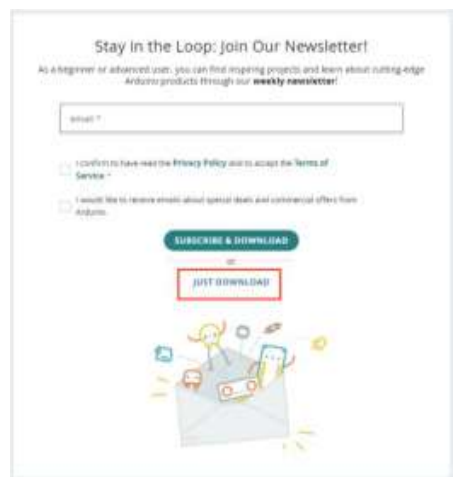
Stáhněte si Arduino IDE pro Mac OS

1) Stáhněte si Arduino IDE

① Stáhněte si webovou adresu Arduino IDE: <https://www.arduino.cc/en/Main/Software>, otevřete adresu URL podle následující tabulky a v budoucnu vyberte odpovídající verzi. Verze softwaru může být aktualizována, stačí nainstalovat nejnovější verzi.



② Vyberte JUST DOWNLOAD.

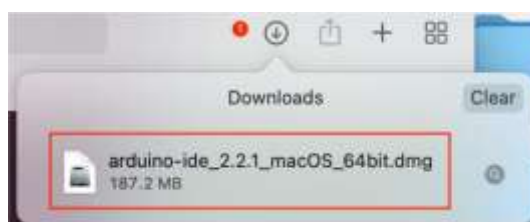


③ Když se objeví následující obrazovka, Arduino IDE se stahuje.



2) Nainstalujte Arduino IDE

① Po dokončení stahování klikněte na ikonu stahování ve vašem prohlížeči a najdete instalační program Arduino IDE.



② Kliknutím na instalační balíček se zobrazí instalační obrazovka, stačí vybrat ikonu Arduino IDE, přejít na Aplikace a program nainstalovat.

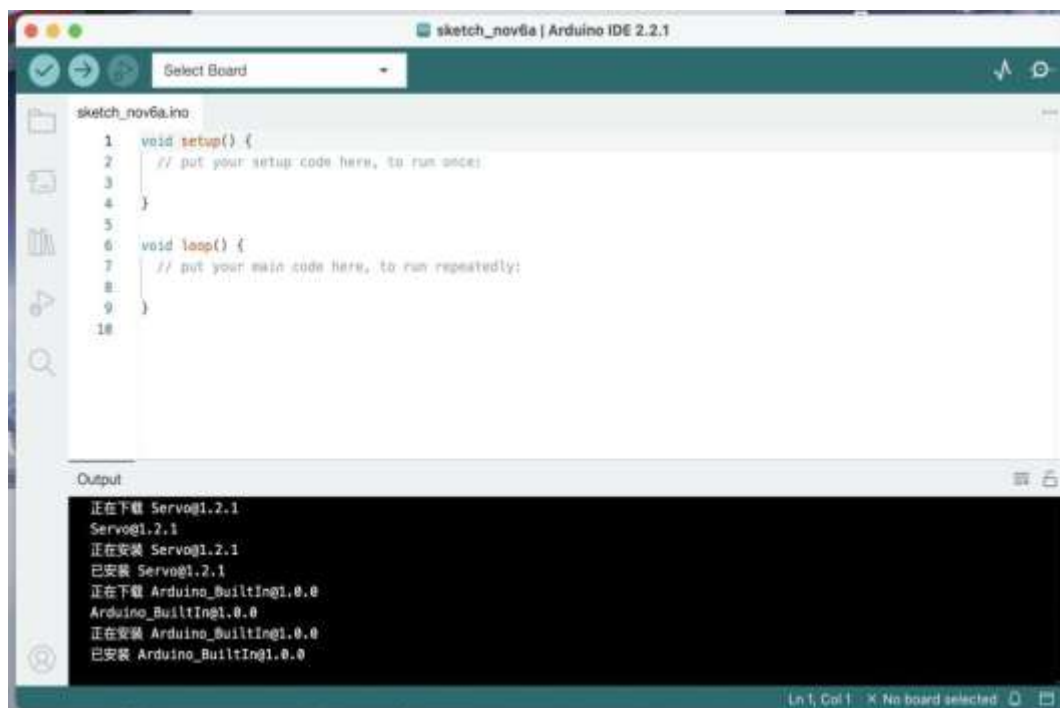


3) Instalace Arduino IDE je dokončena

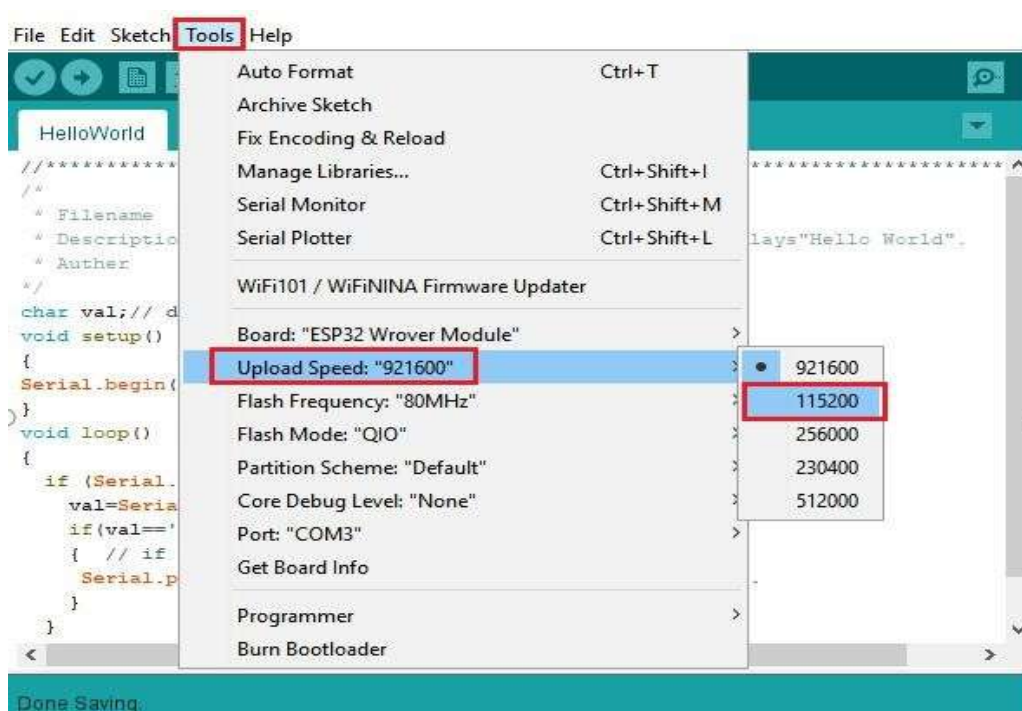
① V launchpadu vyhledejte Arduino IDE a otevřete jej.



② Otevřete program a uvidíte následující rozhraní programu.



③ Pro uživatele Mac OS nastavte baudovou rychlost na 115200 před kliknutím na tlačítko pro nahrání.



Jak nainstalovat sériový ovladač CH340 na MAC

Připojte hlavní řídící desku k počítači pomocí kabelu USB a ovladač se automaticky nainstaluje na systémy MacOS a Windows. Pokud se instalace ovladače nezdaří, musíte ovladač nainstalovat ručně.

1) Stáhněte si ovladače

① Stáhněte si ovladač z webu a rozbalte jej do místního instalačního adresáře. Složku "[CH340 Driver file-mac](#)" naleznete v balíčku zdrojů, který jsme poskytli, toto je soubor ovladače, který chceme nainstalovat.



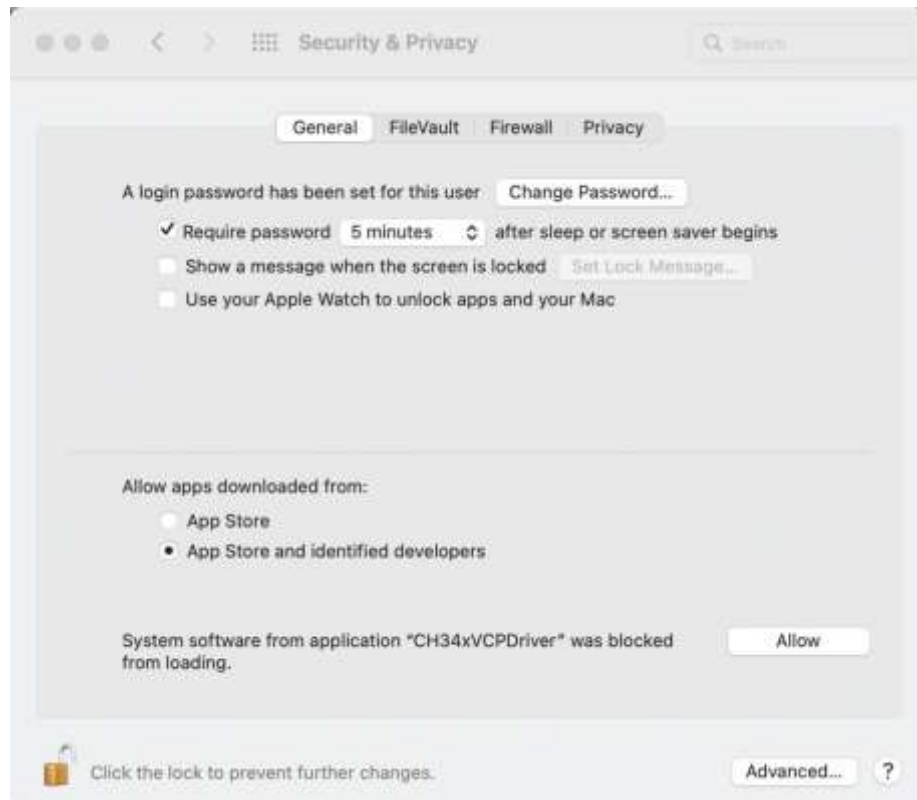
2) Připravte se na instalaci ovladače

① Pokyny pro výchozí instalaci ovladače formátu pkg naleznete v dokumentu s názvem

„Installing the pkg Format Driver“.

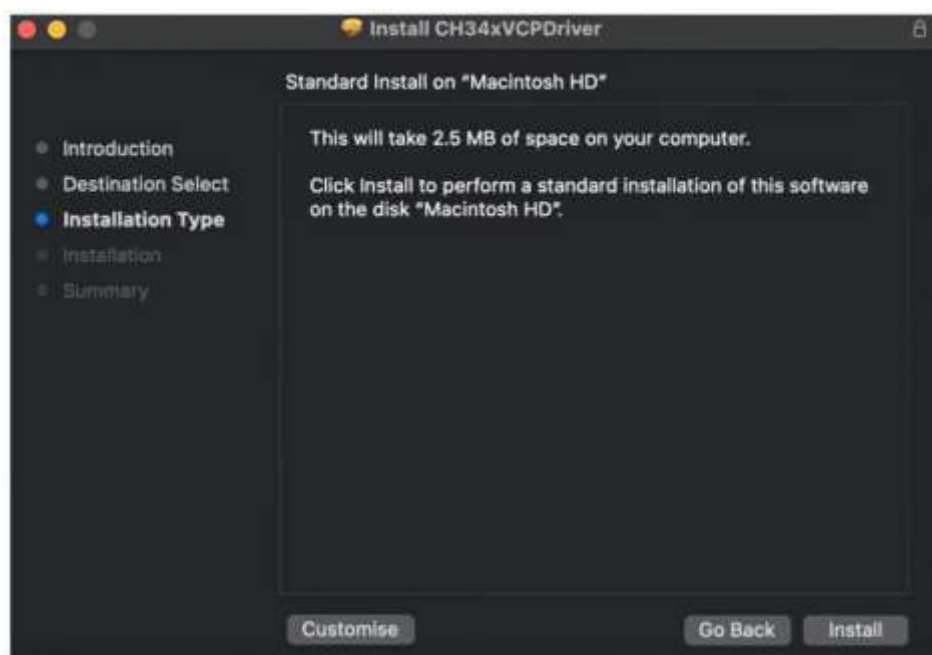
② Pokud Rosetta není podporována v OS X 11.0 nebo novějším, viz "4. Instalace ovladače dmg."

③ Před instalací přejděte na stránku "System Preferences" -> "Security and Privacy" -> "General". V části s názvem „Allow apps downloaded from“ vyberte možnost „Mac App Store and identified developers“, abyste zajistili správnou funkčnost ovladače.

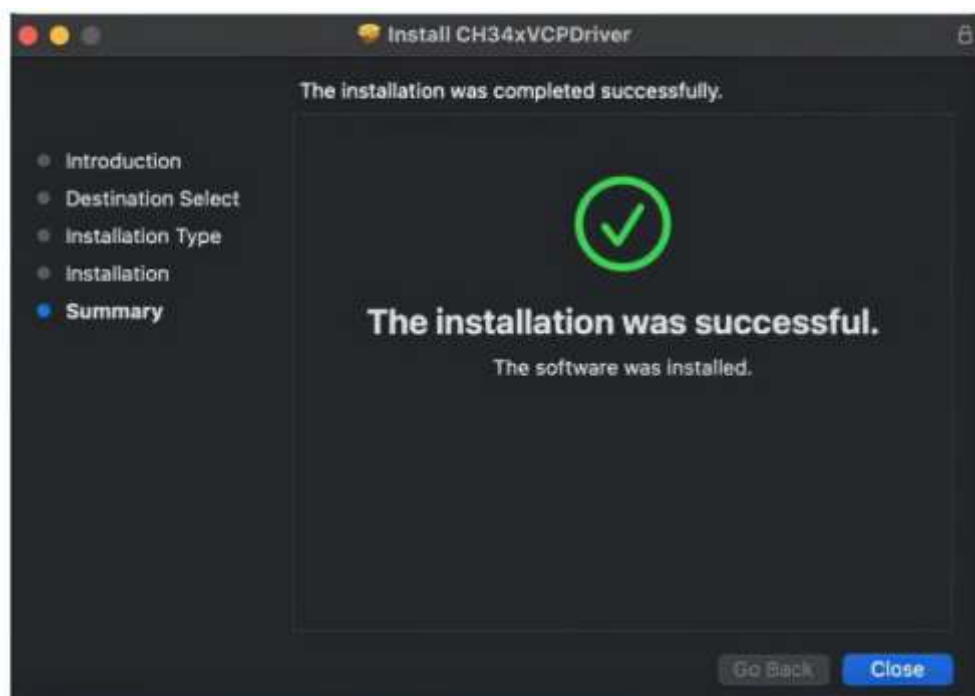


3) Nainstalujte ovladač formátu pkg

- ① Nainstalujte ovladač formátu pkg, klikněte na soubor ovladače -> Continue -> Install.



② Pak úspěšně nainstalujte



- ③ Nainstalujte ovladač formátu pkg na OS X 11.0 a novější: otevřete "LaunchPad" -> "CH34xVCPDriver" -> Install.

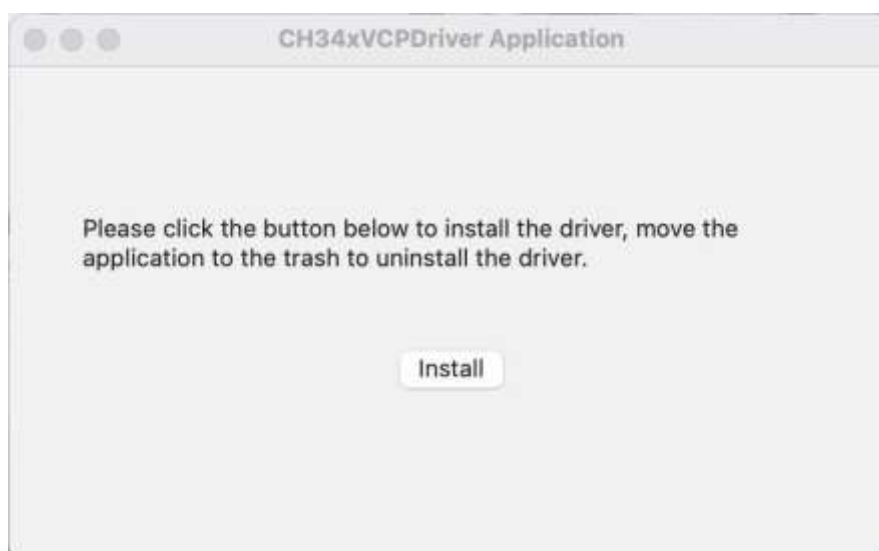


- ④ Pokud používáte OS X 10.9 až OS X 10.15, kliknutím na Restart restartujte počítač a po restartování proveďte následující kroky.



4) Nainstalujte ovladač formátu dmg

- ① Nainstalujte ovladač dmg, klikněte na soubor dmg a přetáhněte „CH34xVCPDriver“ do složky aplikace operačního systému.



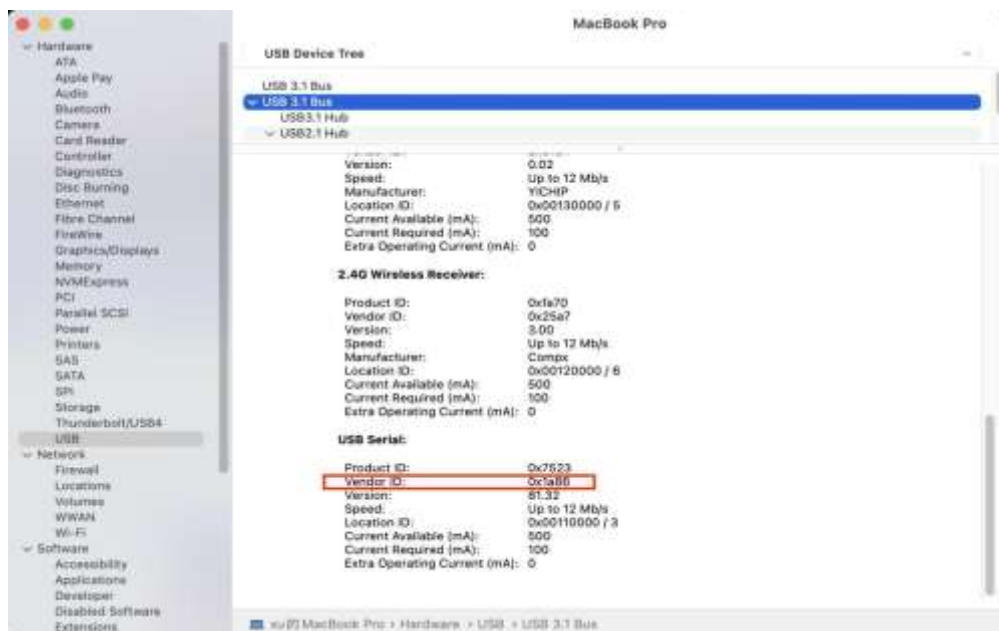
- ② Poté otevřete "LaunchPad" -> "CH34xVCPDriver" -> Install.
- ③ Nakonec se vám objeví obrazovka s nápisem „Install Tips succes“



5) Zkontrolujte, zda je nainstalován ovladač sériového portu CH340

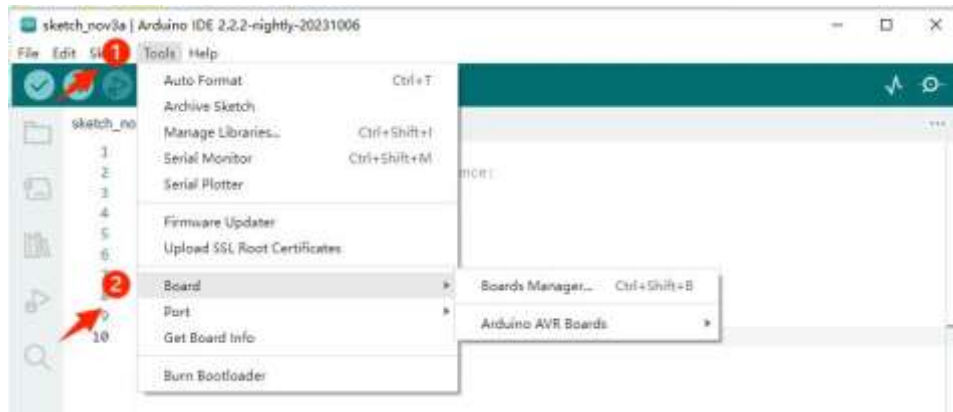
Po zasunutí řídicí desky do portu USB otevřete system report -> Hardware -> USB. Na pravé straně je USB zařízení.

Pokud zařízení USB funguje správně, můžete najít zařízení s "Vendor ID" [0x1a86].



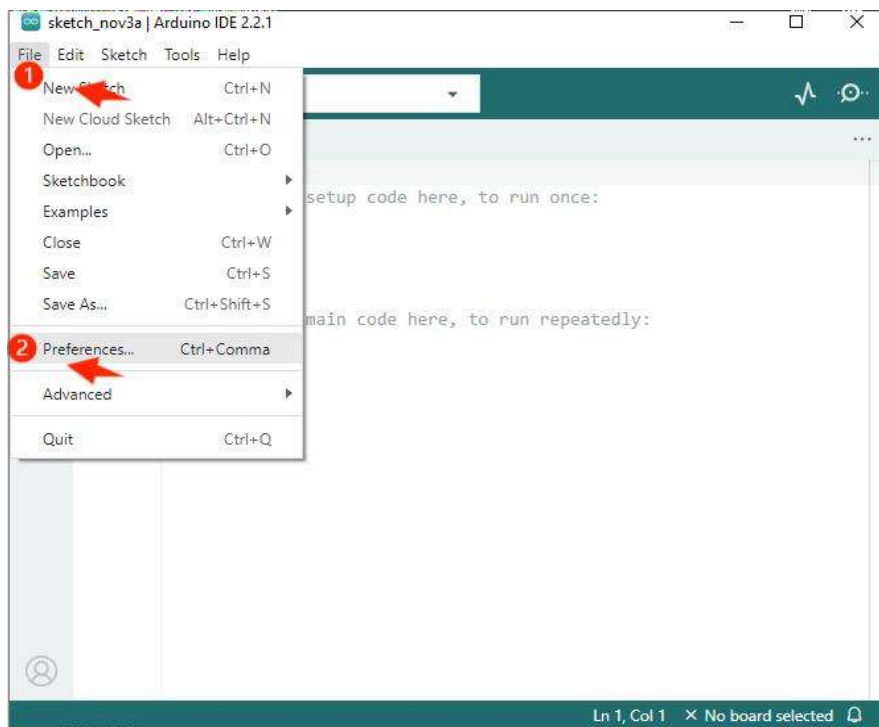
Instalace modulu ESP32 do Arduino IDE

Když otevřete Arduino IDE a vyberete Tools > Board, zjistíte, že v Arduino IDE je pouze Arduino AVR Board a žádný ESP32.

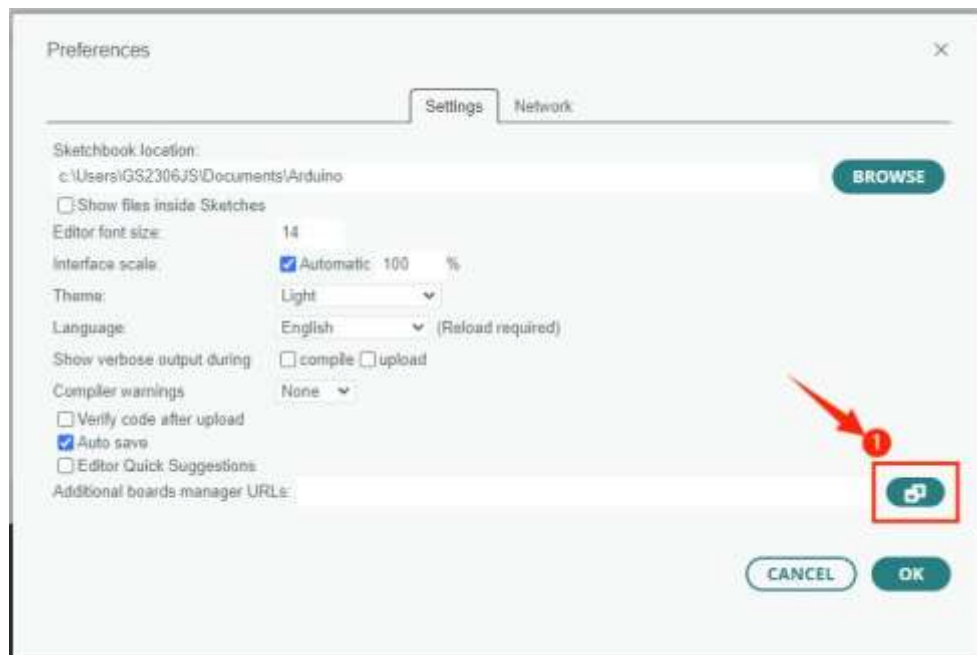


Ovládací deska, kterou tentokrát používáme, je ESP32 ovládací deska, takže musíme nainstalovat ESP32 desku do Arduino IDE. Postupujte podle následujících kroků:

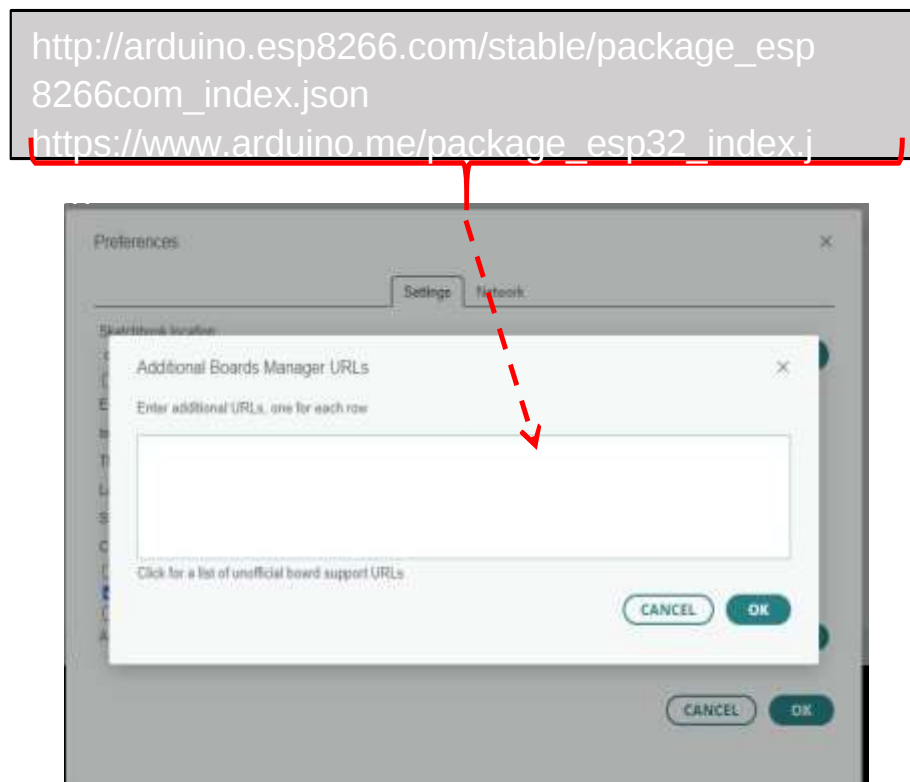
1. Otevřít File > Preferences



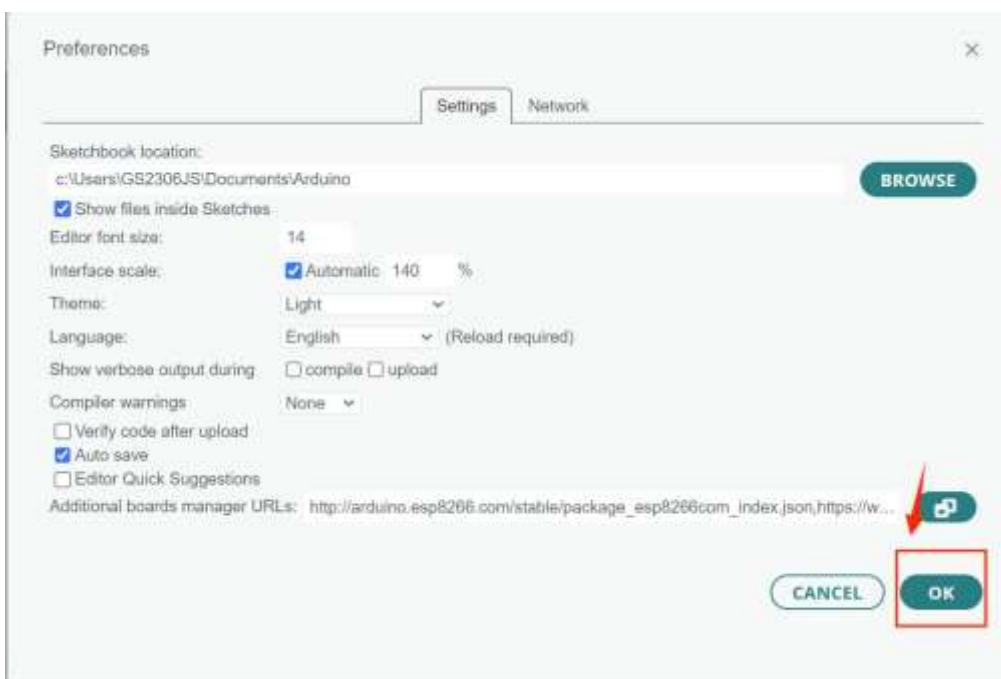
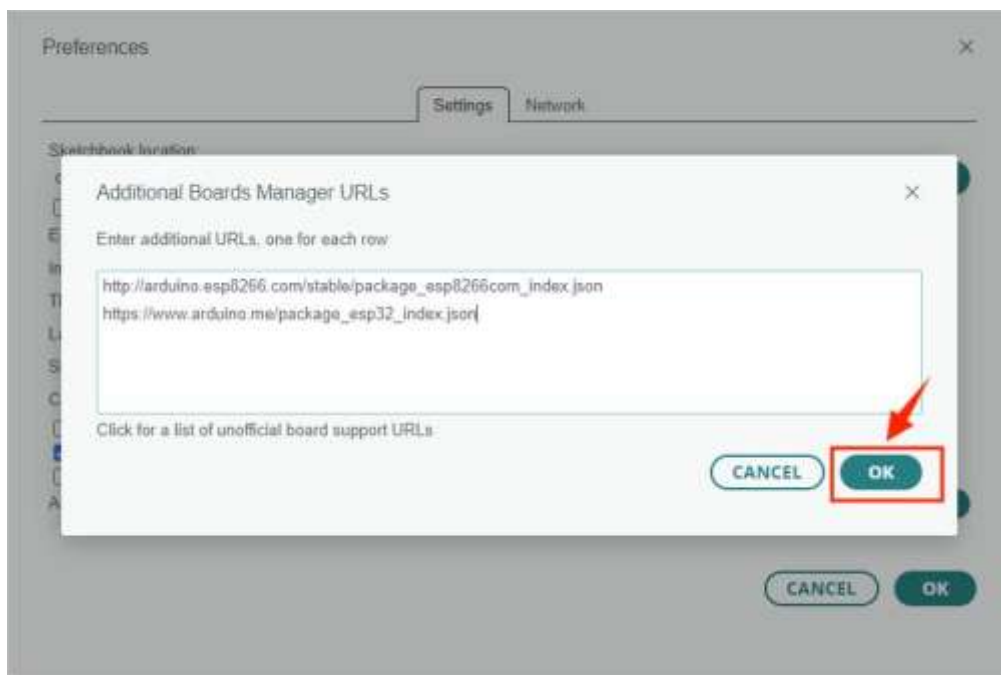
2. Přidat adresu správy vývojové rady URL



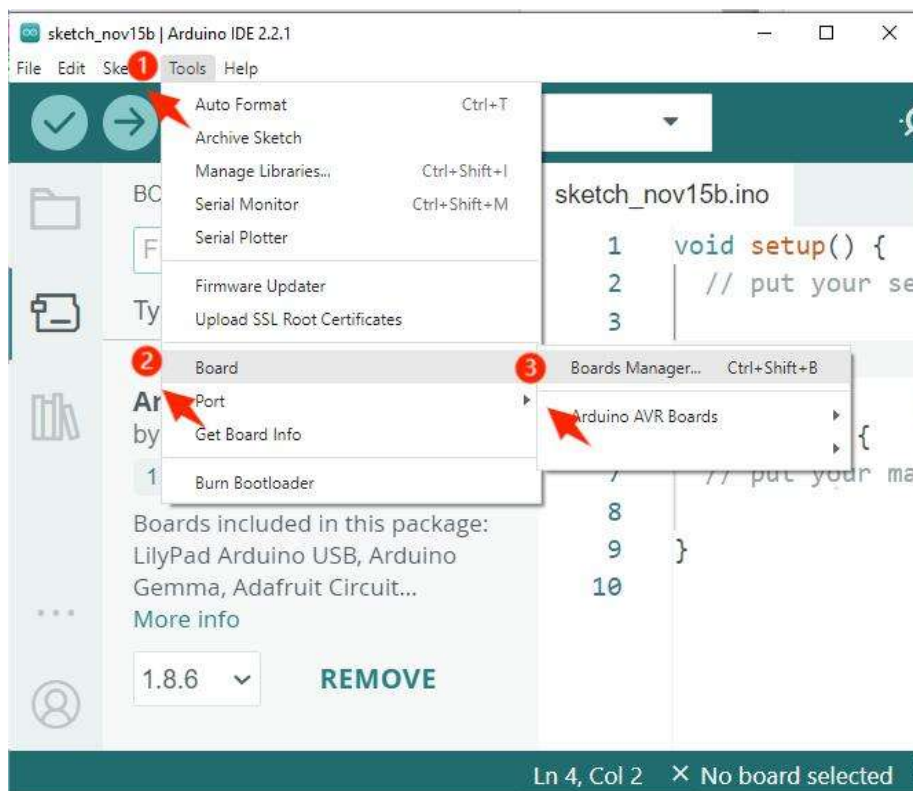
3. Zkopírujte adresu URL z textového pole níže a přidejte ji do "Additional Boards Manager URLs".



4. Po přidání adresy URL klikněte na "OK".

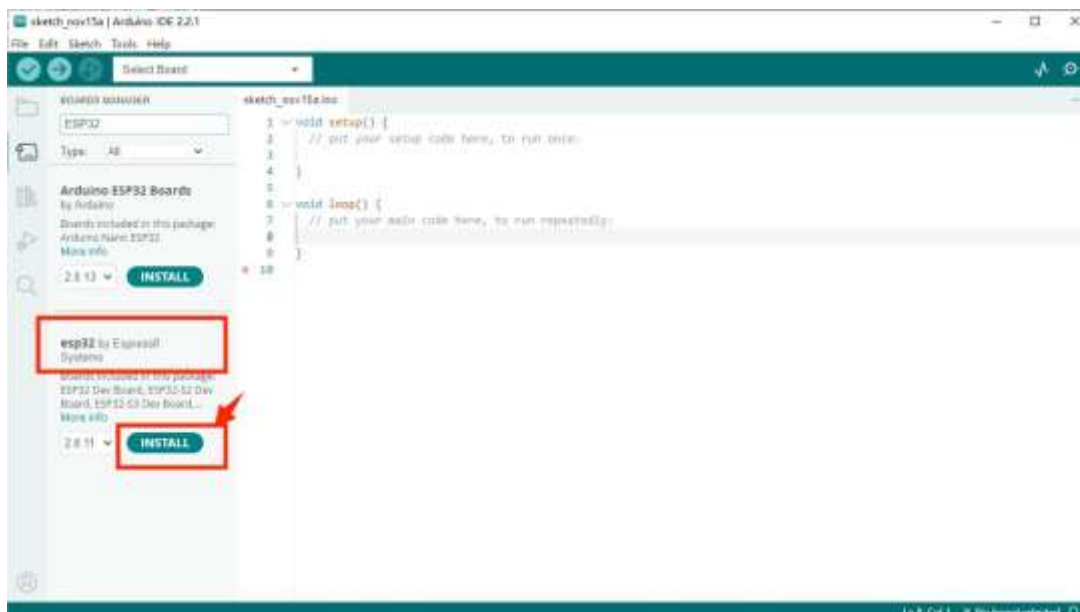


5. Klikněte Tools > Board > Boards Manager...

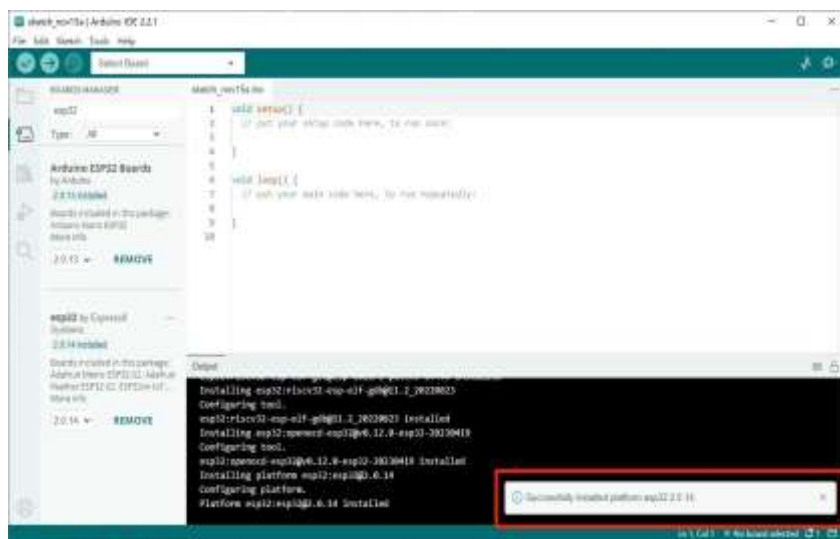


6. Vyhledejte "ESP32" ve vyhledávacím panelu BOARDS MANAGER a nainstalujte ho.

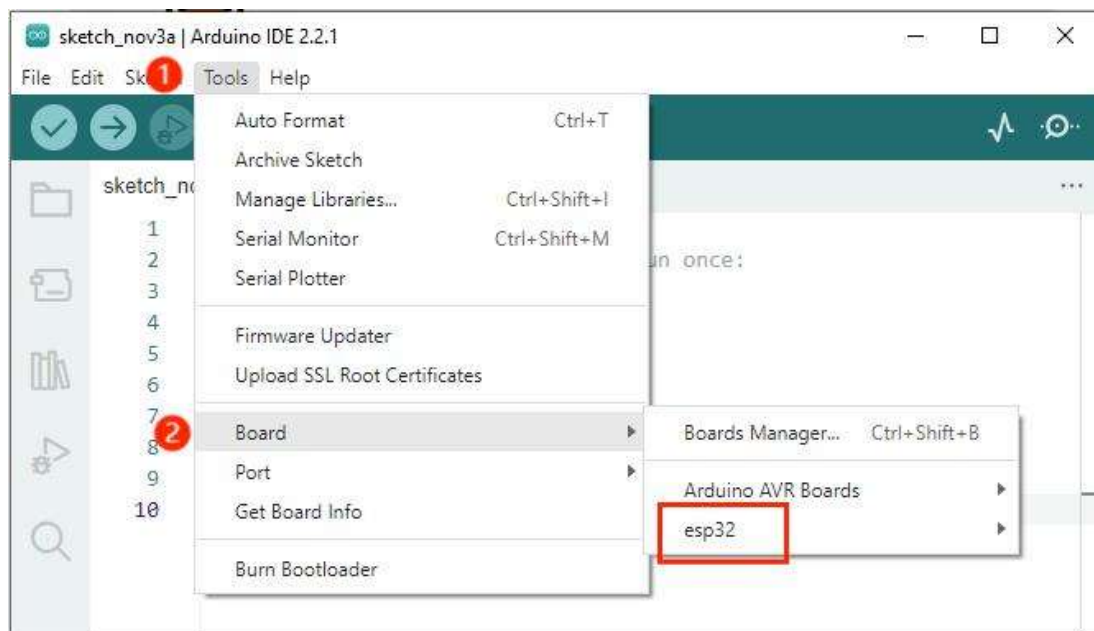
Upozornění: Nainstalujte verzi 2.0.12 esp32, protože nová verze není kompatibilní s knihovnou návodů a může způsobit chyby programu! Pokud byla verze 3.0 již nainstalována, odinstalujte a znovu nainstalujte verzi 2.0 esp32.



7. Počkejte, až bude instalace dokončena a zavřete Arduino IDE, když se zobrazí následující rozhraní.



8. Otevřete Arduino IDE znovu, vyberte Tools > Board a zjistíte, že se objeví deska esp32.



Nyní, když jsme se již seznámili se základním vývojovým prostředím pro desku ESP32, pojd'me ji použít k vytvoření vlastního Inteligentního Dopravního Systému!



Ladění programu a nahrávání



Popis projektu

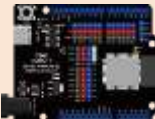


Vyzkoušejte ladění a nahrávání programů, aby se integrovaná LED dioda na ESP32 rozsvítila a zhasla.



Hardwarové schéma

1. Seznam hardware

Obrázek	Název	Množství
	ESP32	1
	Type-C Data Cable	1



Programování

1. Programový blokový diagram



2. Kódování

```
void setup() {  
  //put your setup code here, to run once:  
  //Open the serial port and set the baud rate  
  Serial.begin(115200);  
}  
  
void loop() {  
  //Put your main code here, to run repeatedly:  
  //Print the content you want to display:  
  Serial.println("Hello, Smart Traffic");  
}
```

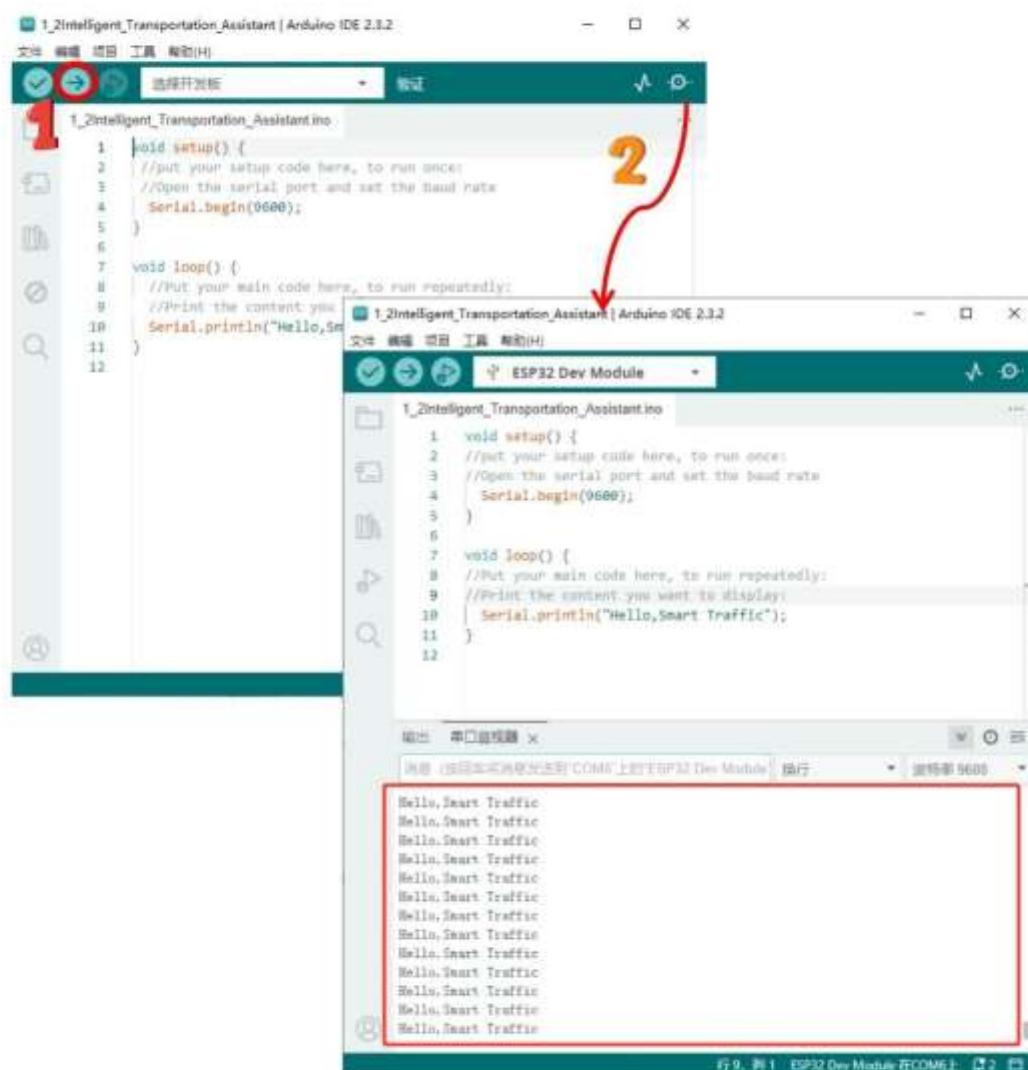
Vysvětlení instrukcí

setup()	Když program začne, instrukce uvnitř funkce setup() budou provedeny jako první. Obvykle se používá pro inicializaci proměnných, I/O portů a některých funkcí. Funkce setup() se spustí při zapnutí nebo resetování řídicí desky a spustí se pouze jednou.
loop()	Funkce loop() se spustí po funkci setup() a instrukce uvnitř ní se mohou vykonávat neustále.
//	Symbol pro komentář na jeden řádek, který se používá k okomentování instrukcí v programu a vysvětlení, jak program funguje. Vše, co je napsáno za symbolem //, nebude kompilováno ani vykonáváno.
Serial	Nástroj pro sériovou komunikaci mezi řídicí deskou a počítačem nebo jinými zařízeními. Obsahuje mnoho instrukcí souvisejících se sériovou komunikací.
Serial.begin(val)	Otevřete sériový komunikační port a nastavte rychlost přenosu dat (baud rate). 'val' označuje hodnotu baud rate.
Serial.println(val)	Toto výstupuje data na sériový port počítače, a data, která se přenášejí, mohou být jakýkoli typ dat. 'val' označuje data, která mají být vytištěna.



Tipy

Když používáme `serial.println()` pro výstup vytištěných dat, můžeme výstup z řídicí desky zobrazit v sériovém monitoru Arduino IDE. Po nahrání programu klikněte na ikonu sériového monitoru, okno sériového monitoru se otevře v rámci IDE a data, která jsou vytištěna sériovým portem, budou zobrazená.



Kapitola 2 Chytré pouliční osvětlení

V nočním městě září pouliční osvětlení jako jasné hvězdy, které nejen osvěcují naši cestu domů, ale také zvyšují bezpečnost na silnicích a krásu města. S neustálým pokrokem technologií se postupně dostává do popředí nový typ inteligentního osvětlovacího zařízení. V této lekci prozkoumáme svět chytrého pouličního osvětlení, porozumíme vědeckým principům a inženýrským technologiím, které za ním stojí, a analyzujeme vliv energie a životního prostředí na návrh pouličního osvětlení.



Sekce 1 Pouliční osvětlení



Cíle kurzu

-  Naučit se základní znalosti o elektrických obvodech
-  Pochopit funkci pouličního osvětlení
-  Seznámit se s historií vývoje pouličního osvětlení a různými typy osvětlovacích technologií
-  Ovládnout používání LED modulů
-  Navrhnout a postavit model pouliční lampy



Znáte funkci pouličního osvětlení? Jaké pouliční lampy se speciálními funkcemi jste již viděli?

Pouliční osvětlení je nezbytnou součástí městské infrastruktury a řízení dopravy. Jeho hlavním účelem je osvětlovat silnice a veřejné prostory, čímž zlepšuje viditelnost a bezpečnost v noci. Pouliční lampy se nacházejí na různých místech, jako jsou ulice, parky, obytné oblasti,



obchodní čtvrti a průmyslové zóny. Některé lampy s výrazným designem také slouží jako prvek pro zkrášlení města a posílení jeho image. S neustálým pokrokem technologií se design a funkčnost pouličního osvětlení rovněž vyvíjí.



Jakým vývojem a evolucí prošlo pouliční osvětlení?

Historie pouličního osvětlení odráží trajektorii urbanizace a technologického pokroku, od nejstarších plynových a olejových lamp až po elektrické lampy v elektrické éře a LED lampy v éře inteligentních technologií.

Olejové lampy: Používaly plyn nebo olej jako palivo a vyžadovaly ruční zapalování a zhasínání.

Klasické elektrické lampy: Mezi běžné typy pouličního osvětlení patřily žárovky, vysokotlaké sodíkové lampy a halogenidové kovové lampy. Žárovky, které byly hojně používány na začátku, jsou neefektivní, produkují hodně tepla a postupně byly vyřazeny. Vysokotlaké sodíkové lampy vyzařují žluté světlo a jsou efektivnější, což je činí nejběžnějším typem pouličního osvětlení v posledních desetiletích. Halogenidové kovové lampy poskytují lepší podání barev a jsou vhodné pro scénáře, kde je vyžadováno vysoké rozpoznávání barev.

LED lampy: Představují nejpokročilejší současnou technologii, jsou vysoce efektivní, úsporné, mají dlouhou životnost, kontrolovatelné rozdělení světla a minimální dopad na životní prostředí.



Víš to?

Vznik elektrických světél

Vznik elektrického světla lze vysledovat až do 19. století, kdy vědci již začali studovat elektrodynamiku a objevili, že světlo může být vytvořeno elektrickým proudem. V roce 1840 britský vědec Frederick de Moleyns vynalezl první elektrické světlo, ale šlo pouze o experimentální zařízení, které nebylo praktické pro každodenní použití. V průběhu následujících desetiletí mnoho vědců pracovalo na vylepšení technologie elektrického osvětlení. V roce 1879 americký vynálezce Thomas Edison úspěšně vyvinul první praktickou žárovku, což znamenalo příchod technologie elektrického osvětlení. Edisonova žárovka byla vyrobena s uhlíkovým vláknem, které vydávalo jasné světlo při zahřátí. Ačkoli měly tyto žárovky krátkou životnost, prokázaly proveditelnost technologie elektrického osvětlení.

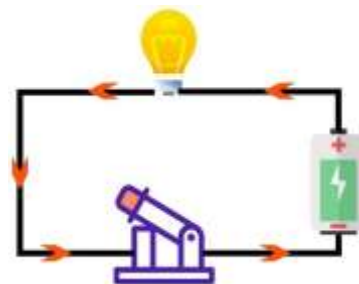


Jak rozsvítit pouliční lampu?



Pro rozsvícení pouliční lampy je potřeba ji umístit do obvodu. Nejzákladnější obvod se skládá z energetického zdroje, elektrického zařízení, vodičů a vypínače (nebo ovládacího zařízení obvodu), které vytvářejí uzavřený okruh.

Když obvod vytvoří uzavřený okruh, elektrický proud začne proudit obvodem. Jakmile proud prochází lampou, lampa přijme elektrickou energii a začne vydávat světlo a fungovat.



Víš to?

Napětí, proud, odpor

Napětí: Napětí je fyzikální veličina, která popisuje potenciální rozdíl mezi dvěma body v obvodu. Obvykle je označováno písmenem "U" a měří se ve voltech (V). Napětí způsobuje pohyb elektrických nábojů z bodu s vyšším potenciálem do bodu s nižším potenciálem v rámci obvodu, čímž vytváří elektrický proud.

Proud: Proud je množství elektrického náboje, který prochází průřezem vodiče za jednotku času. Obvykle je označován písmenem "I" a měří se v ampérech (A). Směr proudu je stejný jako směr pohybu kladných nábojů, tedy od kladného pólu k zápornému pólu.

Odpor: Odpor je míra toho, jak materiál v obvodu brání průchodu elektrického proudu. Obvykle je označován písmenem "R" a měří se v ohmech (Ω). Velikost odporu závisí na vlastnostech materiálu, délce a průřezu vodiče. Při stejném napětí platí, že čím větší je odpor, tím menší je proud, který obvodem prochází.

Vztah mezi těmito třemi veličinami lze popsat Ohmovým zákonem, který říká, že $V = IR$, kde V je napětí, I je proud a R je odpor. Tento vzorec naznačuje, že v obvodu s pevným odporem jsou napětí a proud přímo úměrné. V tomto kurzu se zaměříme na zkoumání, jak využít LED osvětlení k dosažení systémů pouličního osvětlení.

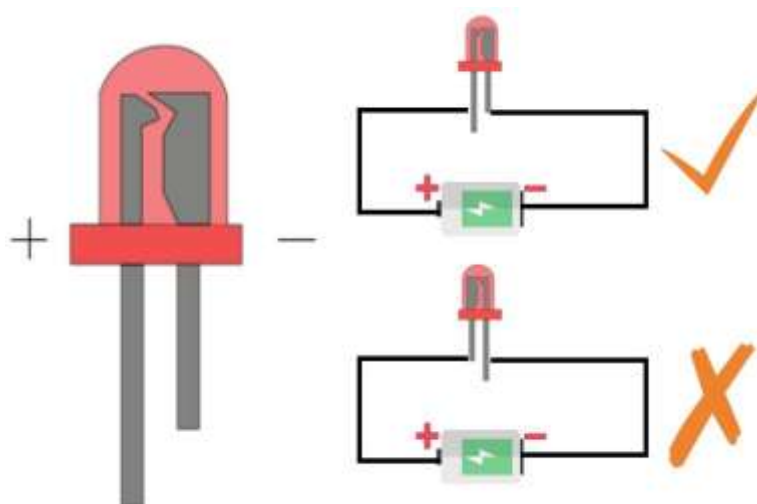
Co je LED?



LED, což je zkratka pro ****Light Emitting Diode**** (světelná dioda), je osvětlovací zařízení vyrobené ze specifických polovodičových materiálů a je široce používáno v oblasti osvětlení. Ve srovnání s žárovkami mají LED diody značné výhody.



Zprvu, jsou energeticky efektivnější a šetrnější k životnímu prostředí, výrazně snižují spotřebu energie a ekologický dopad. Za druhé, LED diody mají extrémně rychlou reakční dobu a dosahují maximálního jasu téměř okamžitě, zatímco žárovky potřebují nějaký čas na zahřátí. Kromě toho lze barvu a jas LED světél přizpůsobit individuálním potřebám, což je činí flexibilnějšími a personalizovanými. Proto má aplikace LED diod v oblasti osvětlení širokou perspektivu. LED světla jsou unidirekční vodiče, což znamená, že mají kladný a záporný pól. Kladný pól LED diody musí být připojen k pozitivnímu výstupu zdroje napájení a záporný pól musí být připojen k zápornému výstupu tohoto zdroje.

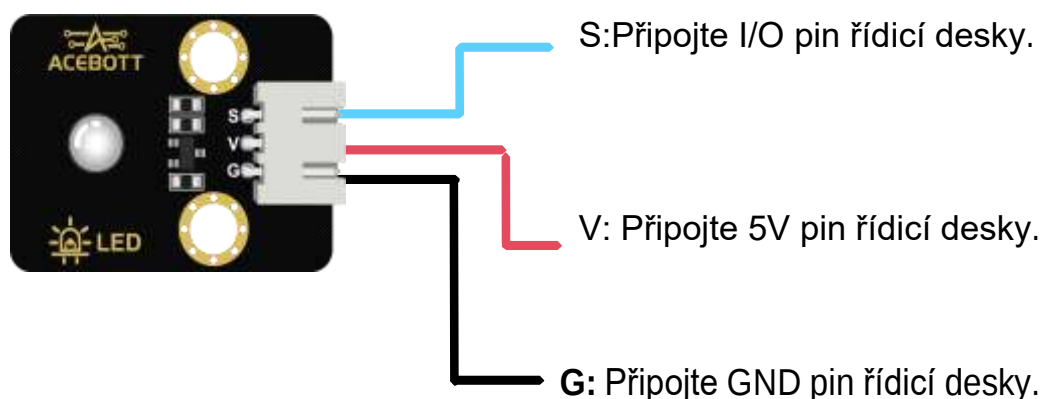


LED Modul



LED modul je integrovaná elektronická součástka s LED diodou. Ve srovnání s jednotlivou LED diodou nám modul usnadňuje její použití a ochranu. Když signální pin (S) LED modulu přijme vysokou úroveň signálu, LED dioda se rozsvítí; když přijme nízkou úroveň signálu, LED dioda se vypne. LED modul má tři piny: S, V a G. Jejich významy a funkce jsou následující:

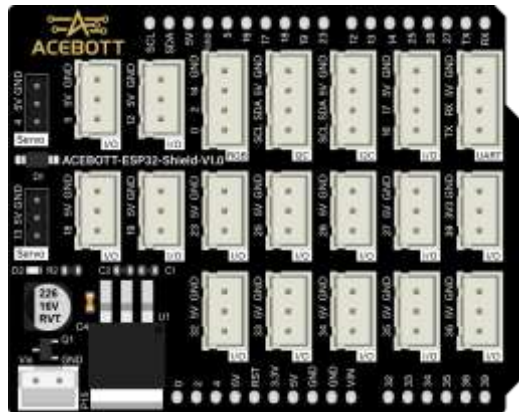
LED modul - Analýza pinů



Rozšiřující deska ESP32



Pro zvýšení pohodlí při používání ESP32 a prevenci problémů, jako jsou chyby v zapojení vedoucí k selhání programu nebo zkratům na řídicí desce, poskytujeme rozšiřující desku pro ESP32. Design rozhraní pro zapojení na této desce je přehlednější,



což činí obsluhu pohodlnější. Když je deska použita spolu s rychlým zapojením, výrazně se snižují chyby v zapojení a efektivně se předchází problémům, jako je nesprávná polarita. Při instalaci rozšiřující desky na ESP32 prosím postupujte podle konkrétních kroků uvedených v manuálu "Assemble Documentation" přiloženém k tomuto učebnímu materiálu.



Rozsvícení pouličního osvětlení



Popis projektu

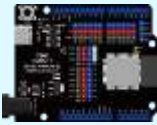
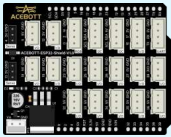
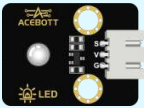


Vytvořte jednoduchý model pouličního osvětlení, ovládejte pouliční světlo, aby se zapnulo a vypnulo, a nechte pouliční světlo blikat v cyklu.

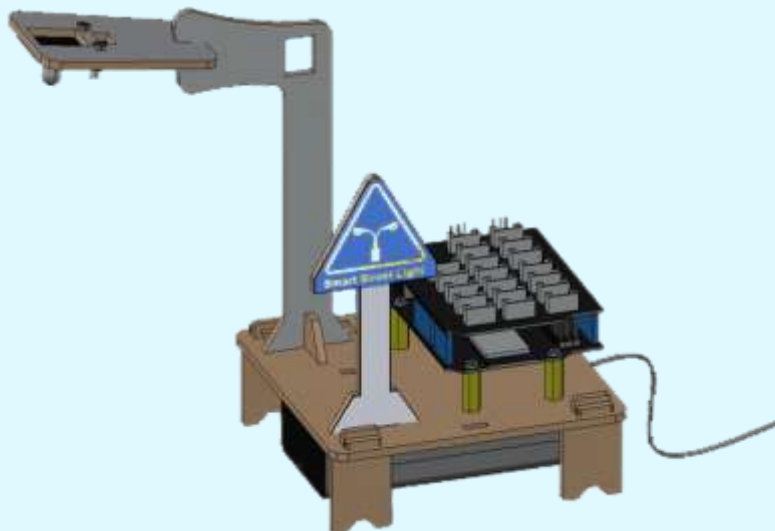


Hardwarové schéma

1. Seznam hardware

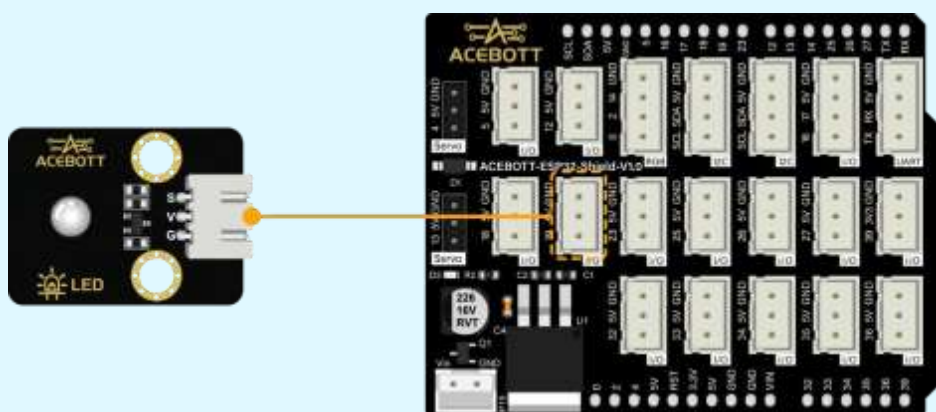
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	White LED	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 1 "01 Ulicí osvětlení" montážního dokumentu.

3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#define LED 19 //Declare the pins of the LED

void setup() {
  pinMode(LED, OUTPUT); //Set LED pins to output mode
}

void loop() {
  digitalWrite(LED, HIGH); //Output high level, light on
  delay(5000);              //Delay 5 seconds
  digitalWrite(LED, LOW);  //Output low level, light off
  delay(1000);             //Delay 1 second
}
```

3. Vysvětlení instrukcí

#define	Před kompilací programu přiřadte konkrétní konstantní hodnotě název, který může být použit v následném programu k reprezentaci této konstantní hodnoty. Tento způsob může zlepšit efektivitu programování a čitelnost kódu.
pinMode(pin,mode)	Nastavte určený I/O pin buď do režimu vstupu, nebo výstupu, kde 'pin' představuje číslo určeného I/O pinu a 'mode' představuje provozní režim, přičemž hodnota je buď INPUT (vstup) nebo OUTPUT (výstup).
digitalWrite(pin,value)	Po nastavení určeného I/O pinu do režimu výstupu můžete tento příkaz použít k nastavení pinu pro výstup digitálního signálu. 'pin' představuje číslo pinu a 'value' představuje hodnotu výstupního digitálního signálu, která může být buď HIGH (vysoká) nebo LOW (nízká).
delay(time)	Pozastaví program na stanovenou dobu, kde 'time' představuje délku pozastavení v milisekundách (ms).



Tipy

- (1) Na desce řadiče ESP32 znamená vysoká úroveň napětí 5V nebo 3.3V, zatímco nízká úroveň znamená napětí 0V. V programu mohou být tyto hodnoty reprezentovány jako „1“ nebo „0“.
- (2) Ne všechny I/O piny na desce řadiče ESP32 mají funkci výstupu. Konkrétně piny 34, 35, 36 a 39 mají pouze funkci vstupu a nepodporují výstup.

Sekce 2: Inteligentní pouliční osvětlení (I)



Cíle kurzu



🚗 Naučit se koncepty digitálních a analogových signálů 🚗

Naučit se pojem proměnných

🚗 Ovládnout způsoby použití proměnných

🚗 Ovládnout použití světelných senzorů

🚗 Navrhnout a postavit inteligentní pouliční osvětlení



Myslíte si, že existují oblasti, kde by se pouliční osvětlení, které jste viděli, dalo zlepšit?

Kdybyste byli požádáni navrhnout inteligentní pouliční osvětlení, jaké inovativní funkce byste přidali? Napište své myšlenky do pole níže.

Design inteligentních pouličních světel by se měl zaměřit na úsporu energie, ochranu životního prostředí, bezpečnost a inteligentní řízení, aby se zvýšila účinnost a kvalita městských osvětlení. Například pouliční světla mohou být vyrobena z obnovitelných materiálů a navržena pro snadnou demontáž, čímž se prodlužuje jejich životnost



a usnadňuje údržba a aktualizace pro dosažení udržitelného rozvoje. Dále mohou být solární panely použity k napájení pouličních světel, využívající solární energii k nabíjení a ukládání energie pro ochranu životního prostředí a úsporu energie. Inteligentní pouliční světla mohou také automaticky upravovat jas na základě změn okolního světla, čímž optimalizují osvětlení a šetří energii.

V této lekci postavíme model inteligentního pouličního světla. Jeho funkcí je využití světelného senzoru k detekci změn v okolním světle a na základě toho dosáhnout automatického řízení pouličního osvětlení.



Senzor světla

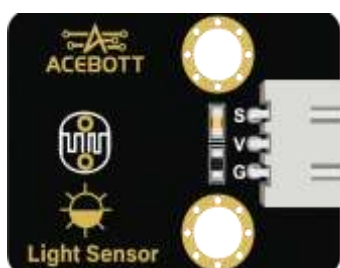


Světelný senzor, známý také jako fotorezistor, je senzor schopný detekovat intenzitu okolního světla.

Jeho princip fungování je založen na odporových vlastnostech fotosenzitivních materiálů, které jsou obvykle vyrobeny z polovodičových materiálů. Odpor světelného senzoru klesá, když je intenzita světla vysoká, a zvyšuje se, když je intenzita světla nízká.

V této lekci světelný senzor, který používáme, se skládá z fotorezistoru, napájecího zdroje a dalších elektronických součástek. Tento senzor převádí detekovanou intenzitu světla na analogovou hodnotu výstupního signálu. Výstupní hodnota je vyšší, když je světlo slabší, a menší, když je světlo jasnější.

Světelný senzor - Analýza pinů



S: Připojte I/O pin řídicí desky.

V: Připojte 5V pin řídicí desky.

G : Připojte GND pin řídicí desky.



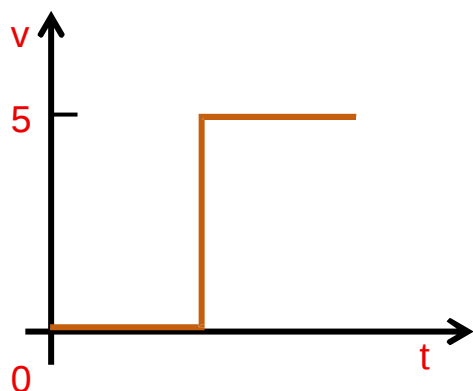
Tipy

V ESP32 jsou analogové vstupní piny pevně stanovené a omezené, přičemž lze použít piny 32, 33, 34, 35, 36, 39 atd. jako analogové vstupní piny, a jejich analogové vstupní hodnoty se pohybují od 0 do 4095.

Digitální a analogové signály



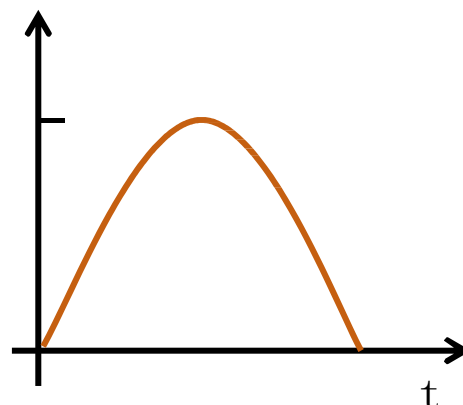
V předchozí lekci jsme použili `digitalWrite()` digitální signál k řízení I/O pinů ESP32, které mohou výstupovat buď 5V nebo 0V napětí, čímž se řídil přepínač světla. Tento typ signálu, který může řídit pouze dvě úrovně napětí, se nazývá digitální signál. Ve skutečnosti se napětí mění kontinuálně, a tento typ signálu, který se neustále mění, se nazývá analogový signál.



Digitální signál je signál reprezentovaný diskrétními hodnotami, který se obvykle používá k přenosu a ukládání informací v elektronických zařízeních nebo komunikačních systémech.

na konečný počet různých hodnot v průběhu času. Například na diagramu vlevo má digitální napěťový signál pouze dvě hodnoty: 5V a 0V.

Analogový signál je signál, který vyjadřuje v informace pomocí kontinuálně se měnících 5 fyzikálních veličin, jako je teplota, vlhkost, jas proud, napětí atd. Analogové signály se často označují jako kontinuální signály, 0 protože mohou mít v určitém časovém rozsahu



nekonečný počet různých hodnot. Například na diagramu vpravo může analogový napěťový signál nabývat hodnot jako 0V, 0.1V, 3.227V, 4.666V, 5V.

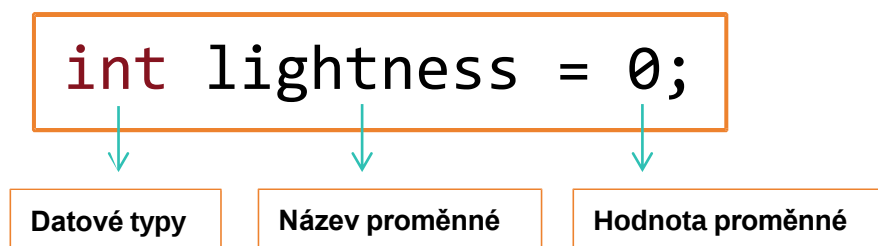
V každodenním životě, jaké signály považujete za digitální signály a analogové signály? Prosím napište své myšlenky na digitální a analogové signály do pole níže.

Díky výše uvedenému učení víme, že výstupní hodnota světelného senzoru je analogový signál, který se může kdykoli měnit. Jak reprezentovat měnící se číselnou hodnotu v programu? Pokud se v programu mění data, potřebujeme použít proměnné k jejich uložení.



Proměnná je kontejner v počítačovém programování, který může uchovávat data, a její hodnota se může měnit kdykoli.

V programování Arduino je syntax pro proměnné následující:

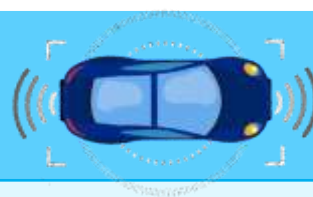


Datové typy: Při vytváření proměnné je nutné nejprve deklarovat datový typ dat uložených v této proměnné. V prostředí programování Arduino základní datové typy zahrnují celé číslo (int), reálné číslo (float), znak (char) a boolean (bool), mezi ostatními.

Název proměnné: Název proměnné reprezentuje význam dat uložených v této proměnné. Prostřednictvím názvu proměnné lze data získat a používat v následných programech.

Hodnota proměnné: Hodnota dat uložená v proměnné. Program používá přiřazovací symbol "=" pro uložení dat do proměnné, a tento symbol "=" může být použit také pro modifikaci dat. Hodnota proměnné může být kdykoli aktualizována a modifikována, aby vyhovovala potřebám fungování programu.

V uvedeném diagramu program definuje proměnnou "lightness" pro reprezentaci hodnoty jasu a přiřazuje jí počáteční hodnotu 0.



Intelligentní pouliční osvětlení (I)



Popis projektu

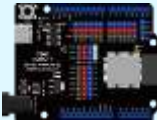
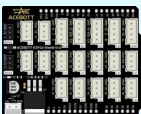


Vytvořte inteligentní pouliční osvětlení, které se automaticky zapne a rozsvítí, když je tma (když je světlo slabé), a automaticky se vypne, když je světlo jasné.

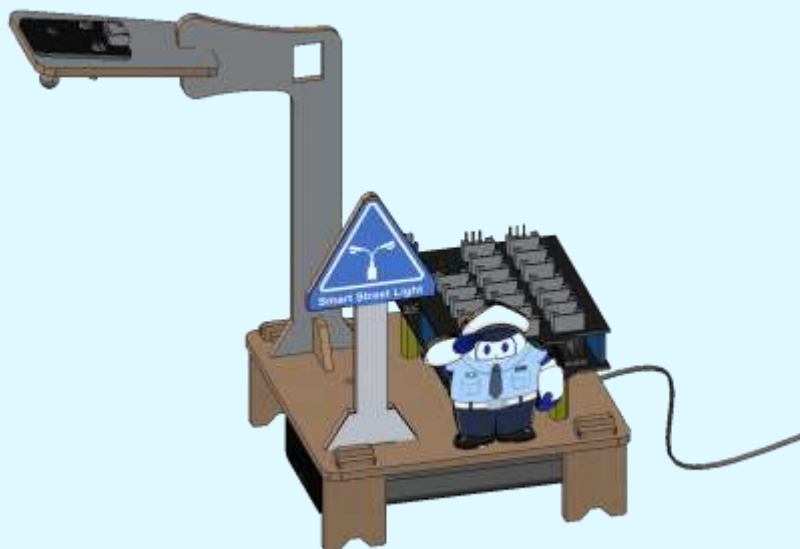


Hardwarové schéma

1. Seznam hardware

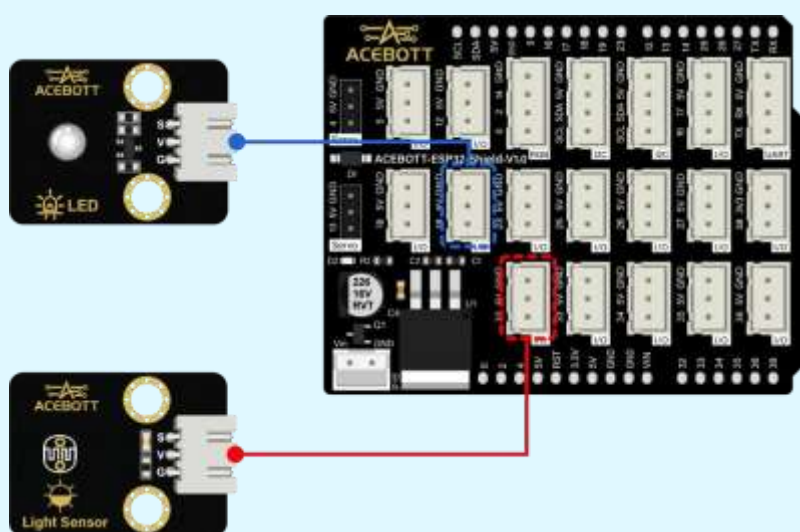
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	White LED	1
	Light Sensor	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 2 "02 Inteligentní pouliční osvětlení (I)" montážního dokumentu.

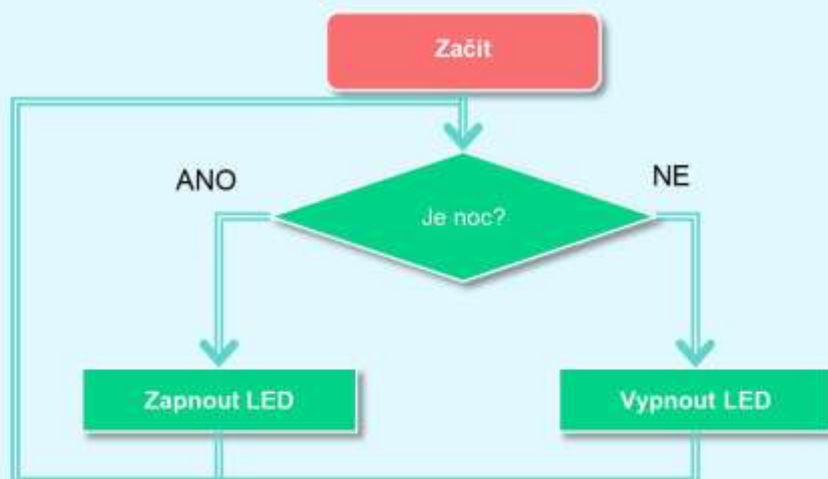
3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#define LED 19 //Declare the pins of the LED
#define PR 32 //Declare the pins of the light sensor

void setup() {
  pinMode(LED, OUTPUT); //Set LED pins to output mode
  pinMode(PR, INPUT); //Set the light sensor pins to input
  Serial.begin(115200); //Initialize baud rate
}

void loop() {
  int brightness = analogRead(PR); //get the light value:
  Serial.println(brightness); //Print light values
  if (brightness > 3000) { //It's night
    digitalWrite(LED, HIGH); //LED on
  } else { //It's daytime
    digitalWrite(LED, LOW); //LED off
  }
}
```

3. Vysvětlení instrukcí

analogRead(pin)	Přečtěte analogové hodnoty ze specifikovaného analogového pinu. ESP32 má více měřicích kanálů, z nichž každý obsahuje 12bitový analogově-digitální převodník, který může mapovat napěťové hodnoty mezi 0–5V na hodnoty mezi 0–4095 pro vstup.
if...else...	Pokud-else příkaz je podmíněná větvení struktura, která se používá v programech pro rozhodování. Jeho syntaxe je následující: <pre>if(Condition Expression){ Statement 1; } else{ Statement 2; }</pre> Když podmínkový výraz vyhodnotí jako pravdivý, provede se příkaz 1; když podmínkový výraz vyhodnotí jako nepravdivý, provede se příkaz 2.

Sekce 3: Inteligentní pouliční osvětlení (II)



Cíle kurzu



-  **Pochopit pojem PWM**
-  **Ovládnout metodu výstupu PWM signálů na ESP32**
-  **Navrhnout a postavit inteligentní pouliční osvětlení**



Jak lépe šetřit energii pro pouliční osvětlení?

V naší poslední lekci jsme diskutovali, jak ovládat zapínání a vypínání pouličních lamp detekcí světla, čímž dosáhneme inteligentního řízení pouličního osvětlení. Nicméně jsme zjistili, že jas těchto lamp zůstává v noci nebo při zatažené obloze

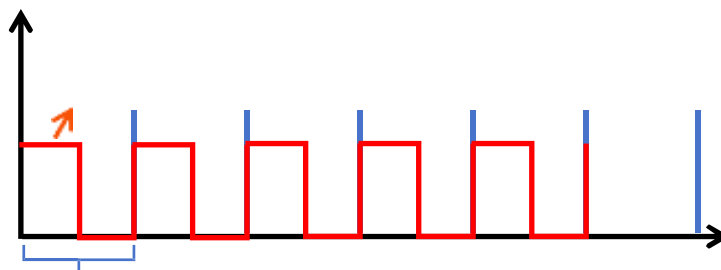


konstantní, což znamená, že většinu času spotřebovávají stejné množství energie. Pro efektivnější úsporu energie můžeme nastavit jas pouličních lamp tak, aby se automaticky přizpůsoboval změnám okolního světla; čím jasnější je okolní světlo, tím slabší je pouliční lampa, a naopak.

Jak tedy můžeme implementovat tento inteligentní řízení jasu? V čipu ESP32 můžeme přesně řídit jas pouličních lamp pomocí modulace šířky impulsu (PWM). Modulace šířky impulsu je technika modulace signálu, která řídí jas světelného zdroje změnou šířky impulsů. V noci nebo při špatných světelných podmínkách mohou pouliční lampy automaticky upravit svůj jas podle okolního světla, čímž šetří energii.



PWM, což je zkratka pro Pulse Width Modulation (modulace šířky impulsu), je technika, která využívá digitální metody k emulaci analogových signálů, aby dosáhla různých úrovní napětí. Přímým řízením připojení (vysoká úroveň) a odpojení (nízká úroveň) obvodů pomocí digitálních signálů a rychlým přepínáním obvodu zapnuto a vypnuto při vysokých frekvencích během určitého časového období lze získat pulzní signál, jak je znázorněno níže.



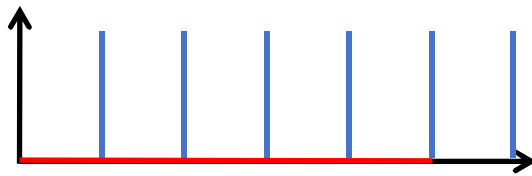
V diagramu uvedeném výše, červená čára představuje pulzní signál.

Vzdálenost mezi modrými čarami označuje periodu pulzního signálu. V každé periodě, doba, po kterou je udržována vysoká úroveň, se nazývá šířka impulsu. Poměr šířky impulsu k periodě je známý jako pracovní cyklus. Technologie modulace šířky impulsu (PWM) dosahuje změny výstupního napětí změnou šířky impulsu.

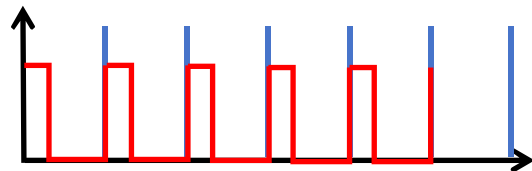
V Arduino můžeme použít `analogWrite(pin, value)` příkaz PWM k výstupu PWM signálu na určený pin. V tomto příkazu parametr v prvním rozbalovacím seznamu představuje číslo pinu, na který chceme výstupní PWM signál, a parametr ve druhém vstupním poli představuje pracovní cyklus PWM signálu. Hodnota vstupu se pohybuje od 0 do 255, což odpovídá pracovnímu cyklu od 0 % do 100 %.

Vezmeme-li jako příklad pin 32 na ESP32, při `analogWrite(32, 0)` použití 0% pracovního cyklu bude výstupní PWM signál mít pracovní cyklus 0 %, což odpovídá výstupnímu napětí 0 V.

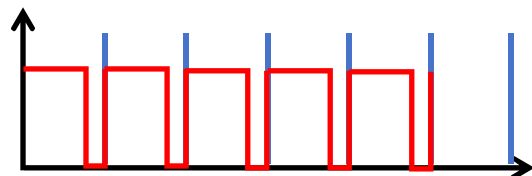
Pokud `analogWrite(32, 127)` použijeme 50% pracovní cyklus, výstupní PWM signál bude mít pracovní cyklus 50 %, což odpovídá přibližně 2,5 V výstupního napětí. Při `analogWrite(32, 255)` použití 100% pracovního cyklu bude PWM signál mít pracovní cyklus 100 %, což odpovídá výstupnímu napětí 5 V.



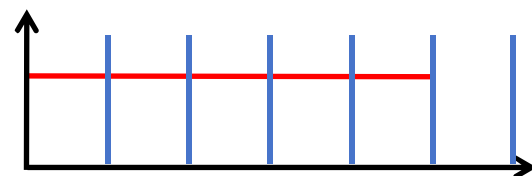
`analogWrite(32,0)`
 0% Pracovní cyklus--0V



`analogWrite(32,64)`
 25% Pracovní cyklus--
 1.25V



`analogWrite(32,191)`
 75% Pracovní cyklus--
 3.75V



`analogWrite(32,255)`
 100% Pracovní cyklus--
 5V



Tipy

Podle specifikací vývojové desky nemají všechny I/O piny schopnost generovat PWM signály. Při využívání PWM výstupu je nezbytné se odkazovat na dokumentaci vývojové desky, aby bylo možné zjistit, které piny mohou generovat

PWM signály. V případě ESP32 kromě pinů 34 až 39, které nemohou generovat PWM signály, všechny ostatní piny schopné výstupu signálu mají funkci PWM výstupu. Tato flexibilita umožňuje uživatelům vybrat si z širokého spektra pinů pro jejich projekty, které vyžadují PWM, což činí ESP32 všestrannou volbou pro různé aplikace



Inteligentní pouliční osvětlení (II)



Popis projektu



Ovládejte jas inteligentních pouličních lamp pomocí světelné hodnoty fotosenzitivních senzorů. Když je světlo slabé, zvýšte jas, a když je světlo silné, snižte jas.

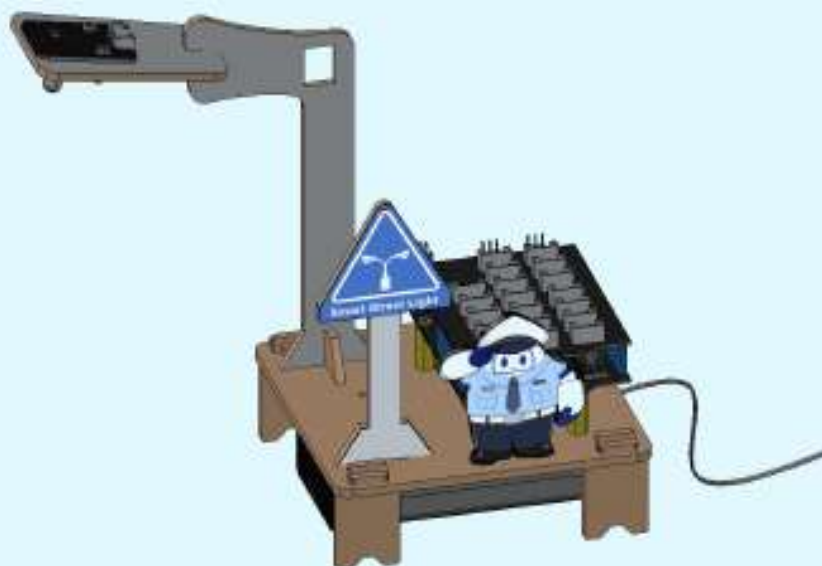


Hardwarové schéma

1. Seznam hardware

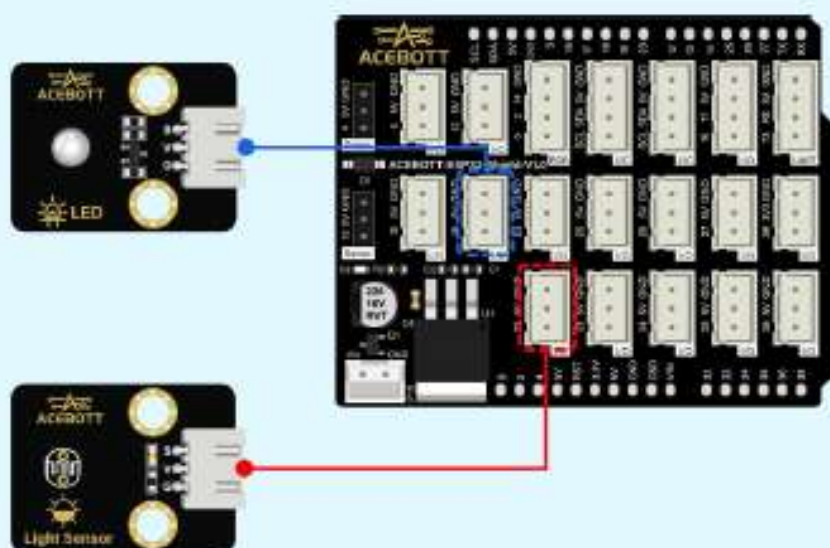
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	White LED	1
	Light Sensor	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 3 "03 Inteligentní pouliční osvětlení (II)" montážního dokumentu.

3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#define LED 19 //Declare the pins of the LED
#define PR 32 //Declare the pins of the light sensor
int brightness;
void setup() {
    pinMode(LED, OUTPUT); //Set LED pins to output mode
    pinMode(PR, INPUT); //Set the light sensor pins to input mode
}
void loop() {
    int light_value = analogRead(PR); //get the lightness value:
    if(light_value < 1500){//check if value is less than 1500
        analogWrite(LED,0);//turn off led
        delay(100); //delay 0.1s
    }
    else{//check if value is more than 1500
        //Convert the light value into an LED brightness value.
        brightness = map(light_value, 1500, 4095, 0, 255);
        analogWrite(LED, brightness); //set LED brightness
        delay(100); //delay 0.1s
    }
}
```

3. Vysvětlení instrukcí

Map
(value,fromLow,fromHigh,toLow,toHigh)

Funkce `map()` doporučně mapuje hodnotu z jednoho číselného rozsahu do jiného. Zde je `value` číslo, které je mapováno, `fromLow` je minimální hodnota původního rozsahu, `fromHigh` je maximální hodnota původního rozsahu, `toLow` je minimální hodnota cílového rozsahu a `toHigh` je maximální hodnota cílového rozsahu.

Kapitola 3: Inteligentní semaforey

V dnešních městech se s rostoucím počtem vozidel ukazuje, že tradiční semaforey nedokážou uspokojit potřeby provozu v době špičky a mimo ni. Objevily se inteligentní semaforey, které využívají pokročilé technologie ke sledování aktuálního provozu a pohybu chodců, inteligentně upravují dobu signálů, optimalizují tok dopravy, zvyšují efektivitu silničního provozu a zajišťují bezpečnost řidičů i chodců.

V této kapitole se budeme zabývat principy fungování a technologickými vlastnostmi inteligentních semaforů a zkoumat jejich klíčovou roli v městské dopravě.



Sekce 1 Základní semaforey



Cíle kurzu



-  Porozumění principu fungování semaforů
-  Porozumění principu modulů semaforů
-  Ovládnutí používání funkcí v programech
-  Vytvoření základního řídicího systému semaforů



Víte, jaké barvy mají semaforey? Co jednotlivé barvy představují?

Semaforey



Semaforey jsou nezbytnou součástí řízení dopravy, používají se k řízení a kontrole plynulosti dopravy, zajišťují bezpečný a pořádný průchod chodců a vozidel křižovatkami. Hlavně indikují směry dopravy a pravidla chování pomocí různých barev světél, což pomáhá snížit dopravní zácpy a míru nehod.

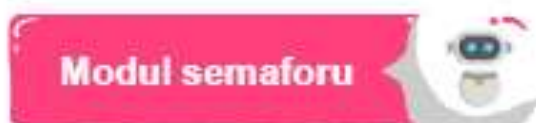


Semaforey se skládají ze tří barev: červené, žluté a zelené. Jejich významy jsou následující:

- **Červené světlo:** Indikuje zastavení, což znamená, že řidiči a chodci musí zastavit na zastávkové čáře nebo čáře pro přechod před semaforem.
- **Žluté světlo:** Indikuje varování, informuje řidiče a chodce, že světlo se chystá změnit na červené, a měli by se připravit na zastavení. Pokud jsou již v křižovatce, měli by ji projít bezpečně co nejdříve.
- **Zelené světlo:** Indikuje pokračování, což umožňuje vozidlům a chodcům bezpečně projít křižovatkou.

Semaforey jsou obvykle řízeny časovačem nebo řídicím systémem na základě plynulosti dopravy. V jednodušších nastaveních se světla mění v pevných časových intervalech. V komplexnějších systémech řízení dopravy mohou světla upravit svůj časový interval a sekvenci na základě reálného dopravního toku, specifických dopravních vzorců během určitých denních hodin nebo zvláštních událostí (například průjezd záchranných vozidel).

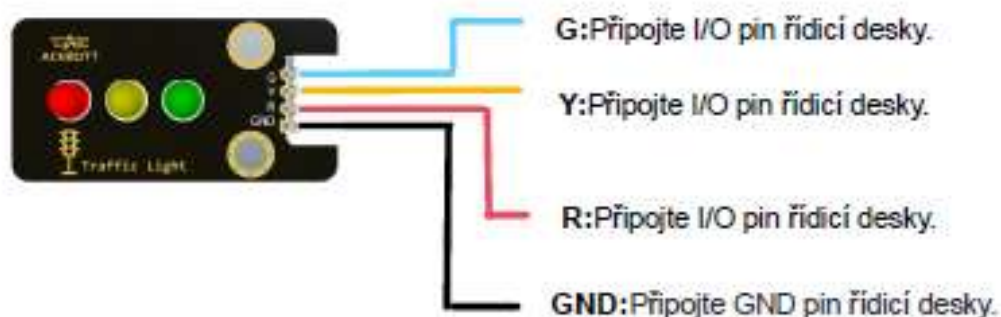
V této lekci budeme používat modul semaforu k simulaci základního systému světelných signálů.



Modul semaforu integruje tři LED diody různých barev: červenou, žlutou a zelenou. Má čtyři piny, z nichž jeden je společný pin, zatímco ostatní tři piny řídí

provoz tří různých barevných LED diod. Společný pin (GND) je připojen k portu GND řídicí desky, zatímco piny R, Y a G jsou připojeny k různým I/O portům řídicí desky. Když každý z pinů R, Y a G obdrží vstup s vysokou úrovní, odpovídající barevná LED dioda se rozsvítí.

Semafor – Analýza pinů





Laboratoř ITS



Základní semaforey



Popis projektu



Simulujte semaforey v reálném životě, dokončete základní systém semaforu a postupně zapínejte zelené světlo na 2 sekundy, žluté světlo na 1 sekundu a červené světlo na 3 sekundy.



Hardwarové schéma

1. Seznam hardware

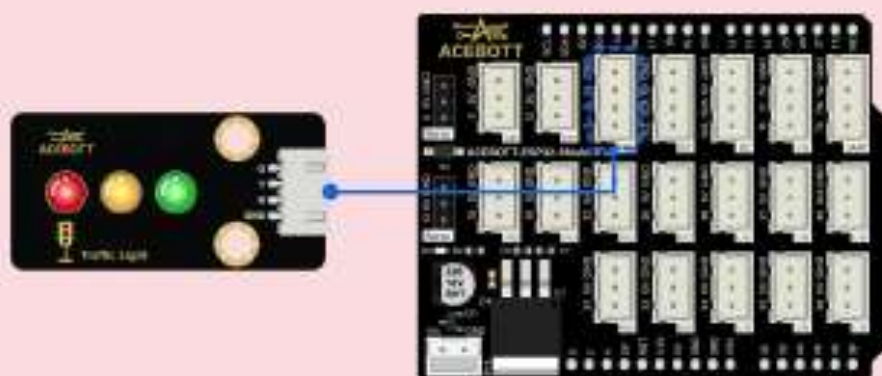
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	Traffic Light Module	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 04 "04 Základní semaforey" montážního dokumentu.

3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#define R 14 //Define pins for red traffic lights
#define Y 2  //Define pins for yellow traffic lights
#define G 0  //Define pins for traffic lights and green lights

void setup() {
    pinMode(R, OUTPUT); //Set the red light pin to output mode
    pinMode(Y, OUTPUT); //Set the yellow light pin to output mode
    pinMode(G, OUTPUT); //Set the green light pin to output mode
}

void loop() {
    Traffic_Light(0, 0, 1); //Green light on
    delay(2000);             //Delay by 2 seconds
    Traffic_Light(0, 1, 0); //Yellow light on
    delay(1000);             //Delay by 1 second
    Traffic_Light(1, 0, 0); //Red light on
    delay(3000);             //Delay of 3 seconds
}

//Define traffic light color control function
//r, y and g are parameters that control the operation of each
//traffic light, with values of 0 or 1, corresponding to LOW and
//HIGH
void Traffic_Light(int r, int y, int g) {
    digitalWrite(R, r);
    digitalWrite(Y, y);
    digitalWrite(G, g);
}
```

3. Vysvětlení instrukcí

Function	Funkce je blok kódu v programu, který vykonává specifický úkol a může být znovu použit. Modularizuje kód, zlepšuje čitelnost programu a efektivitu programování. Funkce jsou rozděleny na knihovní funkce a vlastní funkce. Knihovní funkce jsou vestavěné funkce poskytované základní knihovnou Arduino, jako např. <code>pinMode()</code>, <code>delay()</code> atd. Vlastní funkce jsou vytvářeny uživateli během programování. Syntaxe vlastní funkce je následující: <pre>Return Type Function Name (Parameter List){ Function Body; return Value; }</pre> Return Type: Typ funkce je určen jejím návratovým typem. Pokud funkce vrací celé číslo, návratový typ je <code>int</code>. Pokud funkce vrací číslo s plovoucí desetinnou čárkou, návratový typ je <code>float</code>. Pokud funkce nevrací žádnou hodnotu, návratový typ je <code>void</code>. Function Name: Identifikátor funkce je její název, pomocí kterého lze vyvolat její funkčnost. Parameter List: Používá se k přijímání vstupních parametrů pro funkci. Pokud funkce nevyžaduje vstupní parametry, může být tento část vynechána.
----------	---

Return: Používá se k vrácení specifikovaných dat volajícímu funkce. Typ těchto dat musí odpovídat datovému typu uvedenému před názvem funkce.

Sekce 2 Odpočítávání semaforů



Cíle kurzu



- 🚗 Pochopit funkci časovačů semaforů
- 🚗 Ovládnout použití čtyřmístných digitálních trubic
- 🚗 Ovládnout instalaci knihovny Arduino
- 🚗 Postavit odpočítávací systém semaforů



Viděli jste někdy časovač na semaforu? Jaký je jeho účel?

S pokrokem technologie se funkčnost semaforů neustále vyvíjí. Mezi nimi je funkce odpočítávání široce využívána v moderních semaforech. Semaforey s odpočítáváním jsou inovací v systémech světelných signálů, které přidávají displej s odpočítáváním k tradičním červeným a zeleným světlům, čímž poskytují chodcům a řidičům přesné informace o zbývajícím čase do změny světla. Tento design má za cíl zlepšit bezpečnost a efektivitu silničních křižovatek



a zároveň snížit úzkost a nejistotu u řidičů a chodců.

V této lekci vytvoříme systém semaforu s odpočítáváním přidáním displeje s odpočítáváním k základnímu semaforu a použitím 4místného digitálního displeje k simulaci odpočítávacího časovače.

4-místný trubcový displej



V tomto kurzu používáme modul 4místného digitálního displeje, který se skládá ze čtyř 8-segmentových displejů a dokáže zobrazit více čísel a znaků. Tento modul využívá komunikační protokol I2C k řízení zobrazovaného obsahu, což pomáhá šetřit pinové zdroje na desce řadiče.

4-místný trubcový displej - Analýza pinů



CLK: Připojte pin SCL desky řadiče

DIO: Připojte pin SDA desky řadiče

V: Připojte 5V pin řidičí desky.

G: Připojte 5V pin řidičí desky.

Jak použít komunikaci I2C k ovládání 4místného displeje s trubicí? Abychom mohli pohodlněji ovládat 4místný displej s trubicí, můžeme použít odpovídající knihovnu Arduino pro ovládání modulu.

Arduino knihovna



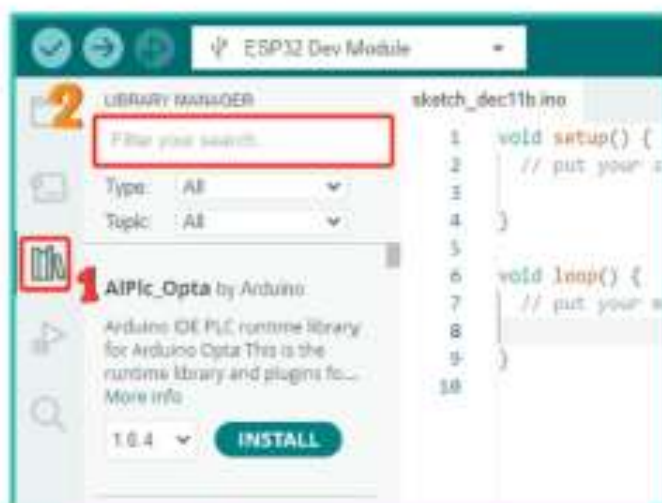
Arduino knihovna je sbírka předem napsaných programů, které umožňují implementaci specifických funkcí voláním funkcí v knihovně. Arduino IDE již

obsahuje mnoho běžně používaných knihoven, jako je knihovna Wire, knihovna Servo atd. Tyto knihovny nám pomáhají rychle vyvíjet různé aplikace. Avšak někdy požadovaná funkčnost není dostupná v předinstalovaných knihovnách, a v takových případech je nutné nainstalovat vlastní knihovny uživatele.

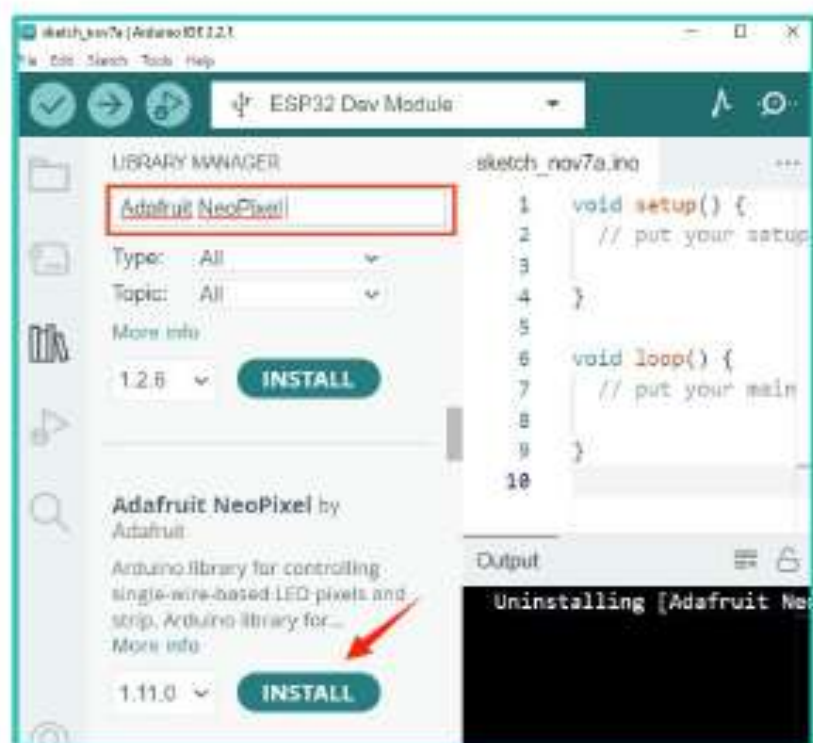
Existují dva způsoby instalace uživatelských knihoven: online instalace a místní instalace.

1. Online instalace

Instalace online knihoven probíhá prostřednictvím Arduino IDE. Protože knihovny jsou stahovány z internetu, vaše zařízení musí být připojeno k síti. Otevřete Arduino IDE, klikněte na ikonu správce knihoven na levé straně, a objeví se vyhledávací pole pro knihovny.

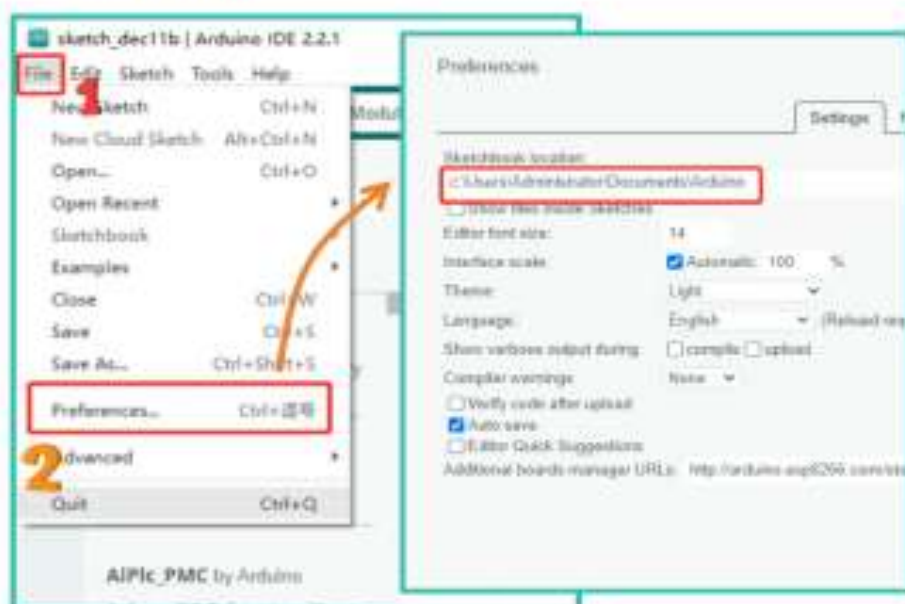


Zadejte název knihovny, kterou chcete nainstalovat, do vyhledávacího pole a po stisknutí klávesy Enter se zobrazí odpovídající výsledky hledání. Najděte požadovanou verzi v seznamu a klikněte na "INSTALL". Po dokončení instalace knihovny je nutné restartovat Arduino IDE, aby bylo možné ji použít.

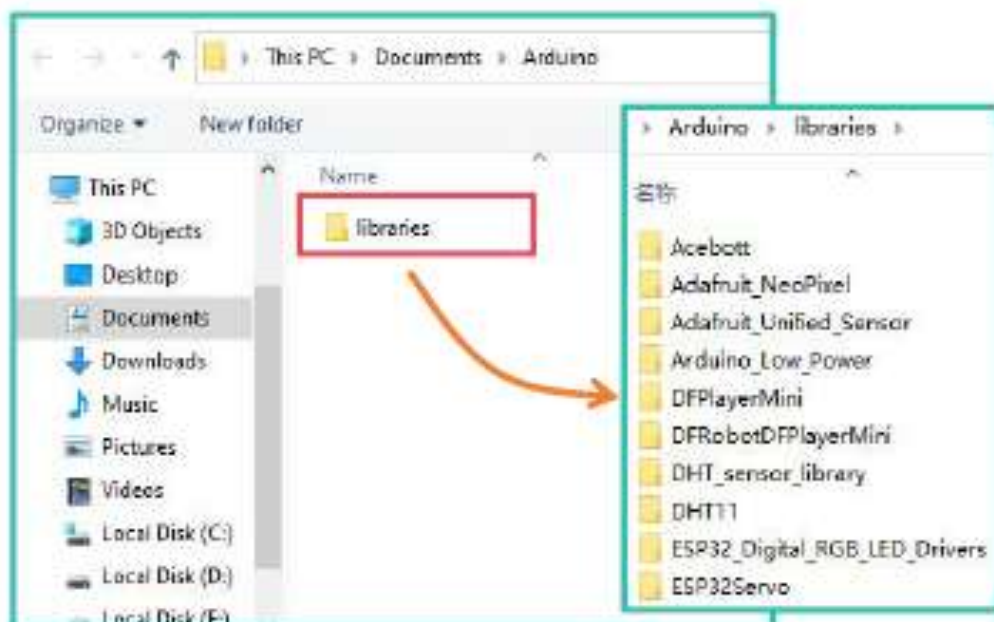


2. Místní instalace

Místní instalace zahrnuje přidání knihovných zdrojů z místního úložiště vašeho počítače do vývojového prostředí Arduino IDE. Arduino IDE má výchozí umístění pro ukládání souborů, a knihovny můžeme jednoduše umístit na toto místo. Výchozí adresu pro úložiště souborů Arduino lze zobrazit v nastavení „Preference“ v Arduino IDE.



Otevřete cestu, kterou jsme si prohlédli výše, a naleznete složku „libraries“ v cestě „Arduino“. Můžete zkopírovat a vložit své knihovní soubory do této složky „libraries“.





Odpočítávací semafony



Popis projektu

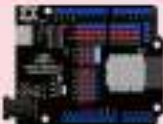


Vytvořte semafor s odpočítáváním a zobrazte zbývající čas každého světla pomocí čtyřmístného digitálního displeje.



Hardwarové schéma

1. Seznam hardware

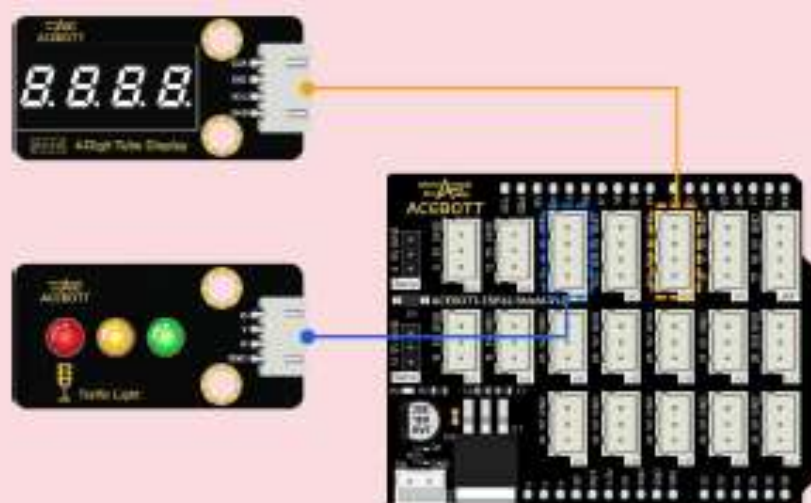
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	Traffic Light module	1
	4-Digit Tube Display	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 05 "05 Semaforey s odpočítáváním" montážního dokumentu.

3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <Wire.h>    //Introducing libraries related to I2C
communication
#include <TM1650.h>  //Introducing 4-digit digital tube related
libraries
#define R 14//Define pins for red traffic lights
#define Y 2//Define pins for yellow traffic lights
#define G 0//Define pins for traffic lights and green lights

char dis[4]; //Create an array to store the displayed strings
TM1650 d;    //Create a TM1650 object
```

```

void setup() {
    Wire.begin();           //Enable I2C communication
    d.init();               //Initialize TM1650 object
    d.setBrightness(TM1650_MIN_BRIGHT); //Set the brightness of TM1650
display
    d.displayOn(); //Turn on display function
    pinMode(R, OUTPUT); //Set the red light pin to output mode
    pinMode(Y, OUTPUT); //Set the yellow light pin to output mode
    pinMode(G, OUTPUT); //Set the green light pin to output mode
}

void loop() {
    Traffic_Light(0, 0, 1); //Green light on
    displayTime(5);         //Delay by 5 seconds
    Traffic_Light(0, 1, 0); //Yellow light on
    displayTime(2);         //Delay by 2 seconds
    Traffic_Light(1, 0, 0); //Red light on
    displayTime(3);         //Delay of 3 seconds
}

//Define traffic light color control function
//r, y and g are parameters that control the operation of each traffic
light, with values of 0 or 1, corresponding to LOW and HIGH
void Traffic_Light(int r, int y, int g) {
    digitalWrite(R, r);
    digitalWrite(Y, y);
    digitalWrite(G, g);
}

//Display countdown function on digital tube
void displayTime(int seconds) { //Countdown display
    //Loop countdown until seconds reach 1
    while (seconds >= 1) {
        itoa(seconds, dis, 10); //Convert numbers into character arrays
        d.displayString(dis); //Display countdown on digital display
        delay(1000); //Refresh the digital display every 1 second
        seconds--; //Seconds minus 1
    }
}
}

```

3. Vysvětlení instrukcí

#include	Používá se k zavedení knihoven mimo program.
Wire.h	Knihovna I2C obsahuje různé funkce související s komunikací I2C.
TM1650.h	Knihovna pro zobrazení na 4-digitální trubici, která obsahuje různé funkce pro ovládání 4-digitálních displejů.
TM1650 d	Vytvoření objektu pro zobrazení na 4-digitální trubici.
d.init()	Inicializační funkce pro zobrazení na 4-digitální trubici.
d.setBrightness()	Nastavte jas zobrazení na 4-digitální trubici.
d.displayOn()	Zapněte funkci zobrazení na 4-digitální trubici.
d.displayString(string)	Zobrazte určený obsah na 4-digitální trubici, kde řetězec představuje řetězec, který má být zobrazen.
while(Condition){ Statement; }	Příkaz "while" je typ smyčky. Nejprve vyhodnotí podmínku uvnitř závorek. Pokud je podmínka pravdivá, vykoná příkazy uvnitř složených závorek a poté znovu vyhodnotí, zda je podmínka stále pravdivá. Pokud je podmínka stále pravdivá, pokračuje ve vykonávání příkazů uvnitř složených závorek. Tento proces pokračuje, dokud podmínka nevrátí hodnotu false, kdy smyčka končí.

Sekce 3 Inteligentní semaforey



Cíle kurzu



- 👉 Pochopení pracovních hodin semaforů
- 👉 Naučit se princip fotocitlivých senzorů
- 👉 Ovládnutí použití fotocitlivých senzorů
- 👉 Navrhnout a postavit systém detekce dopravních prostředků



Je pracovní doba semaforů stejná na každé křižovatce?



Pracovní doba semaforů není na každé křižovatce stejná; je určena podle dopravního toku. Na křižovatkách s vysokým dopravním zatížením je doba zeleného světla pro vozidla relativně delší, aby se zajistil plynulý pohyb vozidel. Naopak, na křižovatkách s menším dopravním zatížením je doba červeného světla pro vozidla delší, aby chodci na přechodech měli více času na bezpečné přejití.

Nicméně, dopravní tok na některých křižovatkách není konstantní; mění se během dne, přičemž některé části dne mají intenzivní dopravu, zatímco jiné jsou klidnější. V takových případech by měla být pracovní doba semaforů nastavena dynamicky, aby odpovídala aktuálním podmínkám.

Abychom vyřešili tento problém, můžeme zavést pokročilou technologii monitorování dopravního toku, která využívá senzory, kamery a další zařízení k monitorování aktuálního pohybu vozidel na silnicích. Na základě těchto reálných dat můžeme upravit dobu trvání světelných signálů,



čímž zajistíme, že časy semaforů budou přizpůsobeny změnám v dopravním toku. Tato optimalizace pomáhá zlepšit nastavení časování světelných signálů.

V této lekci se seznámíme s fotodetektory pro detekci dopravního toku a prozkoumáme, jak je efektivně využívat pro monitorování dopravního toku.

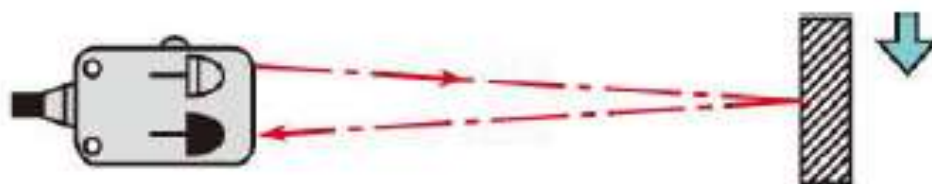
Fotocitlivý senzor



Fotocitlivý senzor vyzařuje „světlo“, například viditelné nebo infračervené světlo, prostřednictvím vysílače a detekuje změny v množství odraženého nebo zablokováného světla detekovaným objektem prostřednictvím přijímače,

čímž získává výstupní signál.

Pracovní princip fotocitlivého senzoru je založen na fotoelektrickém efektu, který přeměňuje energii fotonů na energii elektronickou. Fotocitlivý senzor se skládá z několika částí, včetně světelného zdroje, přijímače a obvodu pro zpracování signálu. Světelný zdroj je primárně používán k osvětlení detekovaného objektu světlem určité intenzity. Přijímač pomocí fotosenzitivního prvku přijímá světelné signály odražené, přenesené, rozptýlené nebo vyzařované měřeným objektem a převádí je na elektrické signály. Nakonec obvod pro zpracování signálu tyto elektrické signály zpracovává zesilováním, filtrováním, porovnáváním a počítáním, aby dosáhl detekce a kontroly objektu.

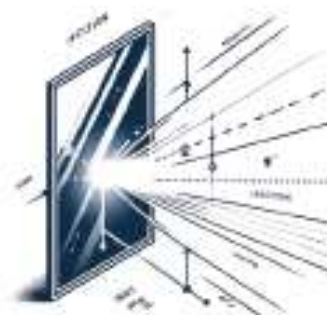




Viš to?

Co je odražené světlo?

Odražené světlo vzniká odrazem světla na rozhraní mezi dvěma různými médii. Když světlo vstoupí z jednoho média do druhého, dochází k odrazu na tomto rozhraní. Úhel odrazu je roven úhlu dopadu a odražené světlo



a dopadající světlo se nacházejí na stejné straně normály. Odraz může být rozdělen na **zrcadlový odraz**, **rozptýlený odraz** a **směrný odraz**, přičemž tyto typy závisí na úhlu světla vůči rozhraní a vlastnostech tohoto rozhraní.

V tomto kurzu jsme použili senzor pro detekci překážek na bázi infračerveného světla jako typ fotocitlivého senzoru. Jeho pracovní princip spočívá v emitování infračerveného světla z vysílacího konce. Pokud je před senzorem překážka, infračervený přijímač detekuje odražené infračervené světlo. V důsledku toho signálový pin vyprodukuje nízké napětí; jinak vydá vysoké napětí.

Infračervený senzor pro vyhýbání se překážkám - Analýza pinů



S: Připojte I/O pin řídicí desky.

V: Připojte 5V pin řídicí desky.

G: Připojte GND pin řídicí desky.



Laboratoř ITS



Intelligentní semaforey



Popis projektu



Použití infračerveného senzoru pro vyhýbání se překážkám k detekci dopravního toku na silnici, kdy při dosažení určité úrovně dopravního objemu prodloužit dobu zelené, aby vozidla mohla projít dále, a zkrátit dobu pro chodce.



Hardwarové schéma

1. Seznam hardware

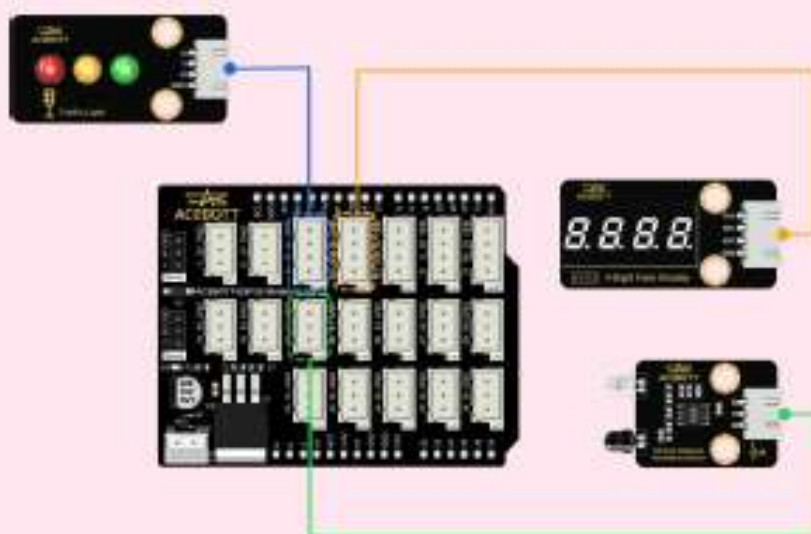
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	Traffic Light	1
	Infrared Obstacle Avoidance Sensor	1
	4-Digit Tube Display	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 06 "06 Inteligentní semaforey" montážního dokumentu.

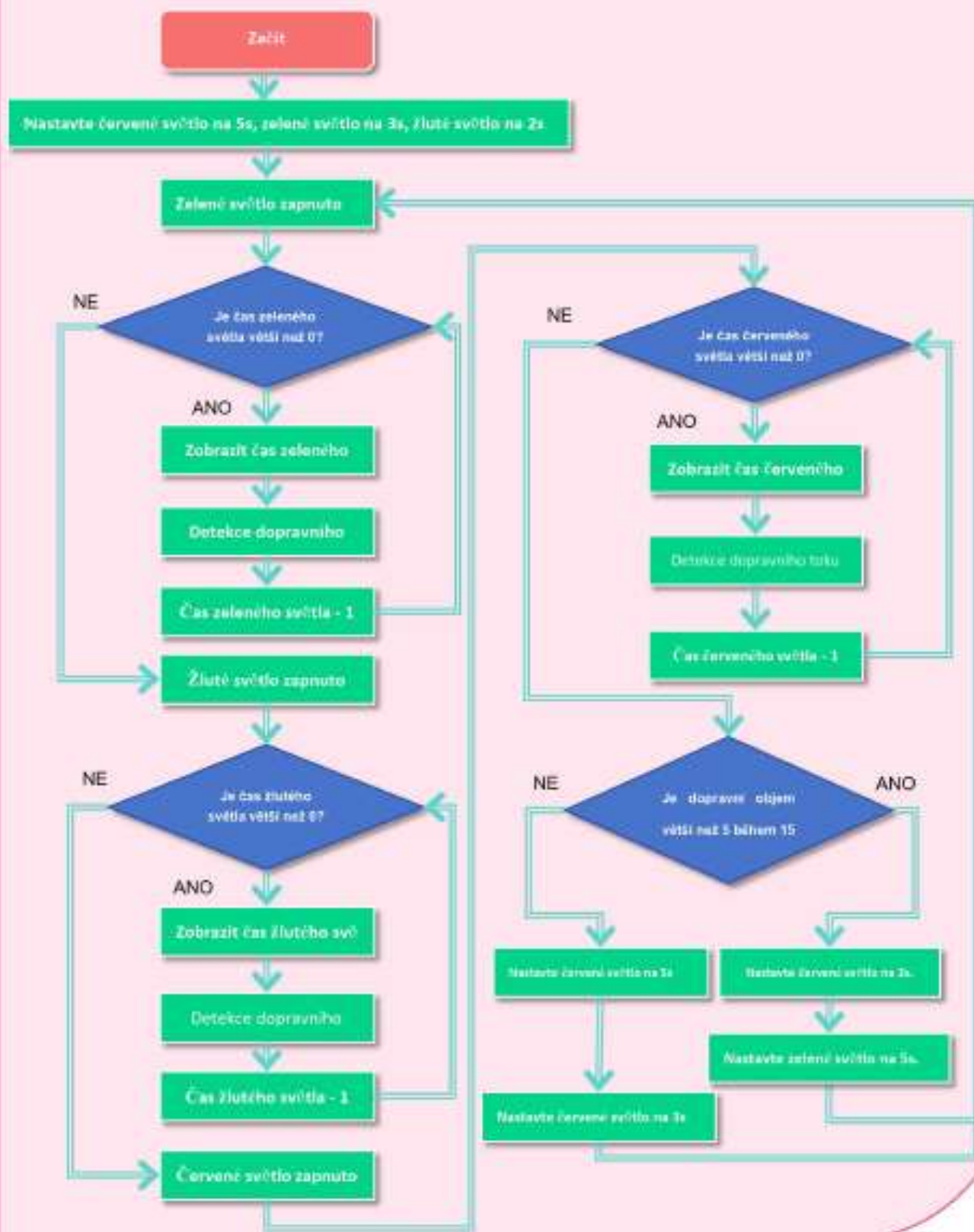
3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <Wire.h> //Introducing libraries related to I2C communication
#include <TM1650.h> //Introducing 4-digit digital tube related
libraries
#define R 14 //Define pins for red traffic lights
#define Y 2 //Define pins for yellow traffic lights
#define G 0 //Define pins for traffic lights and green lights
#define PE 23 //Define the pins of the photoelectric sensor

int carcount = 0;
int redtime = 5000, greentime = 3000, yellowtime = 2000;
//Defining the Time of Traffic Lights
int lasttime = millis();
int Traffic_volume;
int display_time;
char dis[4]; //Create a character array to store the displayed string
TM1650 d; //Create a TM1650 object
char light[3] = { 'g', 'y', 'r' };
char color;
void setup() {
    Wire.begin(); //Enable I2C communication
    d.init(); //Initialize TM1650 object
    d.setBrightness(TM1650_MIN_BRIGHT); //Set the brightness of TM1650
display
    d.displayOn(); //Turn on display function
    pinMode(R, OUTPUT); //Set the red light pin to output mode
    pinMode(Y, OUTPUT); //Set the yellow light pin to output mode
    pinMode(G, OUTPUT); //Set the green light pin to output mode
    pinMode(PE, INPUT); //Set the photoelectric sensor pins to input mode
    Serial.begin(115200); //Initialize baud rate
}
void loop() {
    lasttime = millis();
    for (int i = 0; i < 3; i++) {
        color = light[i];
        trafficlight(color);
        while (display_time >= 1) { //Countdown display
            itoa(display_time, dis, 10); //Convert numbers into character
arrays
```

```

d.displayString(dis); //Display countdown on digital display
for (int j = 0; j < 2; j++) {
    if (digitalRead(PE) == 0) { // Does the infrared obstacle
avoidance sensor detect the vehicle
        carcount += 1;
    }
    delay(500);
}
display_time--; //Seconds minus 1
}
}
//Determine if the traffic flow within 15 seconds is greater than 5
if (carcount > 5 && millis() - lasttime < 15000) {
    redtime = 3000;
    greentime = 5000;
}else{
    redtime = 5000;
    greentime = 3000;
}
carcount = 0;
}
void trafficlight(char color) {
    if (color == 'r') { // Check if the traffic light is red
        digitalWrite(R, HIGH);
        digitalWrite(Y, LOW);
        digitalWrite(G, LOW);
        display_time = redtime / 1000;
    }
    if (color == 'g') { // Check if the traffic light is green
        digitalWrite(R, LOW);
        digitalWrite(Y, LOW);
        digitalWrite(G, HIGH);
        display_time = greentime / 1000;
    }
    if (color == 'y') { // Check if the traffic light is yellow
        digitalWrite(R, LOW);
        digitalWrite(Y, HIGH);
        digitalWrite(G, LOW);
        display_time = yellowtime / 1000;
    }
}
}

```

3. Vysvětlení instrukcí

millis()	Funkce <code>millis()</code> je metoda používaná k získání času, který uplynul od začátku běhu aktuálního programu na desce Arduino. Vrací hodnotu typu <code>unsigned long</code> , která představuje počet milisekund, které uplynuly od začátku běhu programu na desce Arduino až do chvíle, kdy byla funkce zavolána.
Relational operator	Porovnejte velikosti dvou čísel. Pokud je podmínka pravdivá, vraťte <code>true</code> ; jinak vraťte <code>false</code> . <code>a == b</code> : Není <code>a</code> rovno <code>b</code> ? <code>a > b</code> : Je <code>a</code> větší než <code>b</code> ? <code>a < b</code> : Je <code>a</code> menší než <code>b</code> ? <code>a >= b</code> : Je <code>a</code> větší než nebo rovno <code>b</code> ? <code>a <= b</code> : Je <code>a</code> menší než nebo rovno <code>b</code> ?
Logical operator	<code>&&</code> : Výsledek je pravda, pokud jsou obě podmínky pravdivé. <code> </code> : Výsledek je pravda, pokud alespoň jedna podmínka je pravdivá. <code>!</code> : Neguje podmínku; pravda se stane nepravdou a naopak.

Kapitola 4: Chytré přechody

Přechody, jakožto nezbytná součást městských dopravních sítí, nesou důležitou úlohu při zajištění bezpečnosti chodců a podpoře plynulého průběhu dopravy ve městech. V moderní společnosti, s urychleným procesem urbanizace a rostoucí hustotou obyvatelstva, se role přechodů stává stále významnější. Nezahrnují pouze bezpečnost chodců, ale jsou i významným projevem městské dopravní civilizace. V této kapitole se zaměříme na způsoby, jak zlepšit bezpečnost chodců na přechodech a zvýšit efektivitu dopravního provozu.



Sekce 1: Inteligentní ochrana přechodů



Cíle kurzu



- 🚗 Pochopit bezpečnost dopravních přechodů
- 🚗 Naučit se princip fungování servomotorů
- 🚗 Ovládat metody řízení servomotorů
- 🚗 Navrhnout a postavit systém ochrany chodců



Myslíte si, že chodníky jsou dostatečně bezpečné v každodenním životě? Máte nějaké dobré návrhy, jak zlepšit bezpečnost chodníků?

Data ukazují, že dopravní nehody na přechodech se postupně stávají hlavním skrytým nebezpečím v městské dopravní bezpečnosti. V rušných městských ulicích a hustých dopravních sítích je bezpečnost chodců při přecházení vážně ohrožena. Statistiky naznačují, že



každý rok utrpí velké množství chodců zranění nebo dokonce ztrácí životy na

přechodech, což nejen ohrožuje bezpečnost života a majetku jednotlivců, ale také vyvolává širokou společenskou obavu a hlubokou reflexi nad řízením dopravní bezpečnosti.



K redukci výskytu dopravních nehod na přechodech a zajištění bezpečnosti chodců je nutné přiměřeně upravit přechody. Optimalizací struktury přechodu můžeme efektivně zvýšit úroveň dopravní bezpečnosti a snížit pravděpodobnost nehod. Možnosti

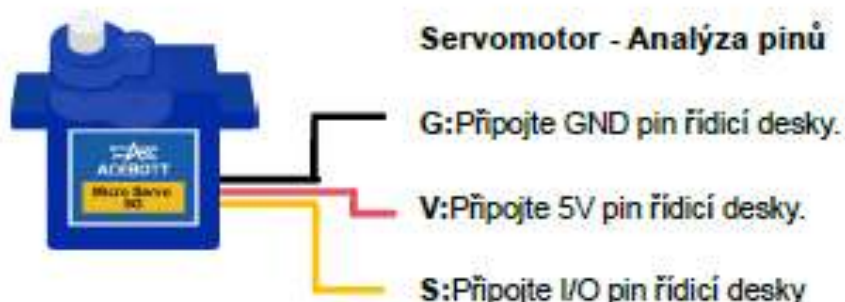
jako rozšíření přechodu, instalace zábran a jasné vymezení chodcovských a nechodcovských zón mohou být implementovány k zajištění bezpečných interakcí mezi vozidly a chodci.

V této lekci postavíme bezpečný přechod pomocí servotechnologie k vytvoření inteligentní zábrany, která efektivně oddělí vozidla a chodce.



Servomotor je běžný typ motoru, který dokáže řídit rotaci motoru na konkrétní úhlovou pozici. Uvnitř servomotoru se nachází motor, zařízení pro zpětnou vazbu polohy (obvykle potenciometr) a řídicí obvod.

Servomotory jsou obvykle řízeny pomocí signálů Pulse Width Modulation (PWM). Šířka pulzu PWM signálu určuje úhel rotace servomotoru. Kontinuálním upravováním šířky pulzu PWM signálu lze servomotor řídit tak, aby rotoval v určitém rozsahu úhlů, čímž se dosáhne přesné kontroly polohy. Servomotory se běžně používají v aplikacích, jako je robotika, dálkově řízená vozidla a modely letadel, které vyžadují přesnou kontrolu polohy. Model servomotoru, který používáme v této lekci, je 180stupňový SG90 servomotorový modul, s rozsahem pohybu od 0 do 180 stupňů.





Inteligentní ochrana chodníků



Popis projektu



Použijte servomotor k řízení otevírání a zavírání zábradlí. Když je světlo v pruhu červené, zábradlí se spustí, aby oddělilo vozidla a chodce. Když je světlo zelené, zábradlí se zvedne a vozidla projedou.



Hardwarové schéma

1. Seznam hardware

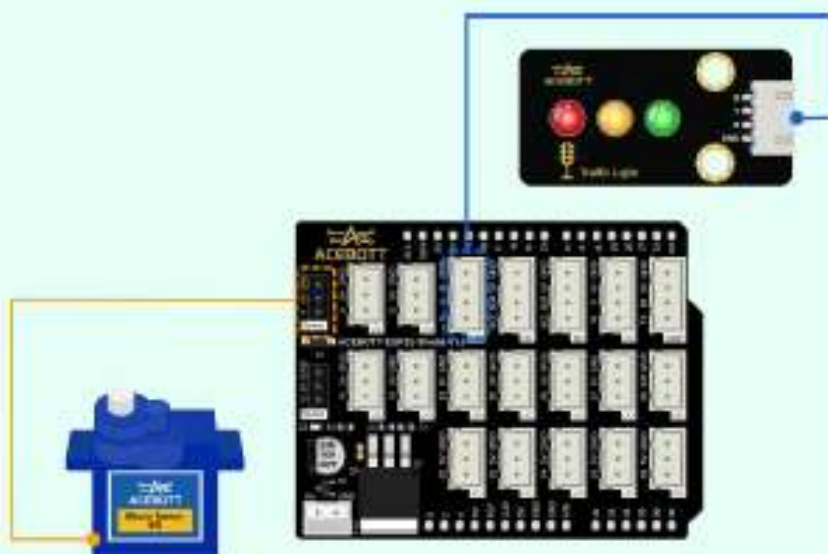
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	Servo	1
	Traffic Light	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 07 "07 Inteligentní bezpečnostní ochrana pro přechod" montážního dokumentu.

3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <ESP32_Servo.h> //Calling the servo program library
#define R 14 //Define pins for red traffic lights
#define Y 2 //Define pins for yellow traffic lights
#define G 0 //Define pins for traffic lights and green lights
#define serve 4 //Declare the pins of the servo motor
Servo MyServo; //Generate an object for a servo motor
const int railing_close = 90; //Configure the angle for servo railing closure
const int railing_open = 0; //Configure servo railing opening angle

void setup() {
    pinMode(R, OUTPUT); //Set the red light pin to output mode
    pinMode(Y, OUTPUT); //Set the yellow light pin to output mode
    pinMode(G, OUTPUT); //Set the green light pin to output mode
    MyServo.attach(serve, 500, 2500); //Initialize servo
    MyServo.write(railing_close); //Configure servo initial position
    delay(500);
}

void loop() {
    Traffic_Light(0, 0, 1); //Green light on
    MyServo.write(railing_open); //Railing open
    delay(2000); //Delay 2 seconds
    Traffic_Light(0, 1, 0); //Yellow light on
    delay(1000); //Delay 1 second
    Traffic_Light(1, 0, 0); //Red light on
    MyServo.write(railing_close); //Railing closed
    delay(3000); //Delay 3 seconds
}

//Define traffic light color control function
void Traffic_Light(int r, int y, int g) {
    digitalWrite(R, r);
    digitalWrite(Y, y);
    digitalWrite(G, g);
}
```

3. Vysvětlení instrukcí

Servo.h	Hlavičkový soubor servomotoru
Servo MyServo	Vytvoříte objekt servomotoru obsahující odpovídající parametry a související operace servomotoru.
MyServo.attach(Pin,Min,Max);	Inicializujte servomotor Pin: Odkazuje na pin, který propojuje servomotor s hlavní řídící deskou. Min: Reprezentuje minimální šířku pulzu servomotoru. Max: Reprezentuje maximální šířku pulzu servomotoru.
MyServo.write()	Řídit servomotor, aby se pohyboval do specifikované úhlové pozice.

Sekce 2: Pozemní dopravní světla



Cíle kurzu



- 👤 Pochopit pozemní dopravní světla
- 👤 Naučit se princip fungování světelných pásků
- 👤 Ovládat metody řízení světelného pásku
- 👤 Navrhnout a postavit pozemní dopravní světla pro chodce



„Jaké jsou způsoby, jak zvýšit povědomí o bezpečnosti chodců při průchodu po chodnicích?“

Chodci, jako důležití účastníci silničního provozu, nemohou být při řešení bezpečnostních otázek opomíjeni. Musíme však čelit nepříjemné skutečnosti, že mnoho dopravních nehod souvisí s chováním chodců na silnici. K zlepšení této situace lze přijmout několik opatření ke zvýšení bezpečnosti chodců na přechodech:



Vzdělávání a informování veřejnosti: Organizování různých informačních kampaní k šíření významu pravidel pro chodce a bezpečnostního povědomí, čímž se zvýší vědomí a iniciativa chodců dodržovat dopravní předpisy.

Dopravní značení: Zvýšení počtu značek pro chodce, dopravních značek a upozornění na křižovatky, které chodcům připomínají, aby věnovali pozornost dopravním úsekům a signálům, čímž se snižuje pravděpodobnost nehod.

Zlepšení dopravních zařízení: Zlepšení designu přechodů instalací zpomalovacích prahů, bariér a pozemních světel pro chodce, které chodce vedou k dodržování předpisů a zvyšují jejich povědomí o bezpečnosti na silnicích a sebeochraně.

Dále se zaměříme na diskusi o tom, jak instalace pozemních světel na přechodech může zvýšit povědomí o bezpečnosti.

Pozemní světlo



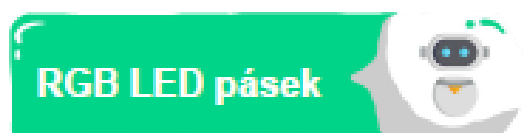
Pozemní dopravní světla jsou navržena především k tomu, aby přitahovala pozornost chodců, kteří se dívají na své telefony. S rozšířeným používáním chytrých telefonů se 'telefonní zombíci' stali bezpečnostní hrozbou v městské dopravě. Pozemní dopravní světla poskytují jasné

vizuální signály pod úrovní očí chodců, což je účinně upozorňuje na aktuální stav dopravního signálu.

Zlepšení viditelnosti v noci: V málo osvětlených prostředích mohou osvětlená světla na zemi účinněji upoutat pozornost chodců, zejména když je viditelnost omezená nebo je špatné počasí.

Kombinace pozemních a vertikálních dopravních světel může komplexně zlepšit bezpečnost chodců na přechodech. Pozemní dopravní světla doplňují možné mrtvé úhly vertikálních světel, zejména když se chodci dívají dolů nebo se pohybují v přeplněných podmínkách.

V této lekci použijeme RGB LED pásy k simulaci pozemních dopravních světel na přechodech, což studentům pomůže lépe pochopit pracovní principy a účinky dopravních signálů intuitivním způsobem.



RGB LED pásy se skládá z několika RGB LED diod, přičemž každá RGB LED dioda obsahuje červenou, zelenou a modrou LED. RGB LED pásy umožňují variace v počtu LED, barvách a úrovních jasu, což umožňuje vytváření různých barevných efektů.



Viš to?

Tři základní barvy světla

Mícháním těchto tří základních barev světla v různých intenzitách lze vytvořit jakoukoliv barvu světla. Tento způsob míchání se nazývá aditivní míchání. Například, když se smíchají červené a zelené světlo se stejnou intenzitou, vytvoří žluté světlo. Pokud se současně přidá modré světlo, může to vytvořit další barvy, jako je oranžová, fialová a azurová.



Princip RGB základních barev byl široce aplikován v mnoha oblastech, zejména v televizi, počítačových displejích, barevném osvětlení a digitální fotografii.

V tomto kurzu používáme modul WS2812 LED pásu, který je tvořen několika RGB LED diodami propojenými do série. Každá RGB LED je vybavena samostatným řídícím čipem. Tyto řídící čipy přijímají a odesílají data sériově, což umožňuje nezávislé řízení jasu a barvy každé LED diody na pásu. Na platformě Arduino existuje několik knihoven pro řízení LED pásů, které nám usnadňují jejich ovládání. V tomto kurzu používáme knihovnu `Adafruit_NeoPixel` pro řízení RGB LED pásu.

RGB LED pásek - Analýza pinů



DIN: Připojte I/O pin řídící desky.

+5V: Připojte 5V pin řídící desky.

GND: Připojte GND pin řídící desky.



Laboratoř ITS



Pozemní dopravní světla



Popis projektu



Pro synchronizované pozemní dopravní světlo: Když je dopravní světlo červené, pozemní světlo svítí zeleně pro chodce, a když je zelené, pozemní světlo svítí červeně, signalizující zastavení.

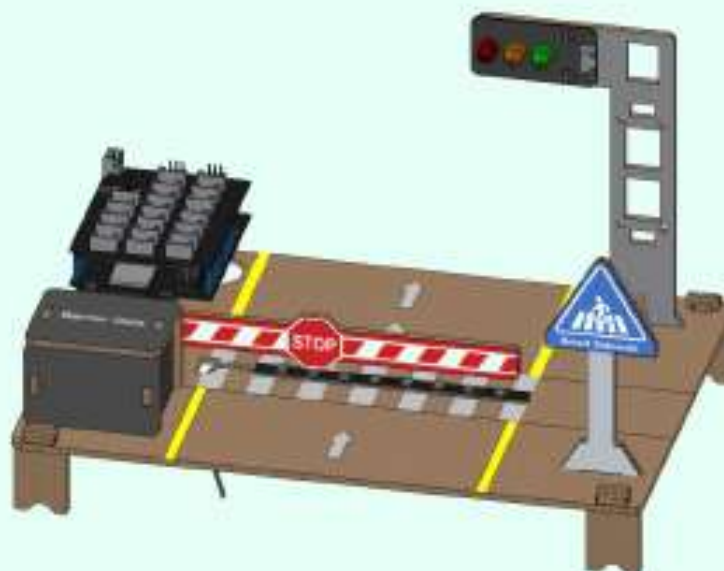


Hardwarové schéma

1. Seznam hardware

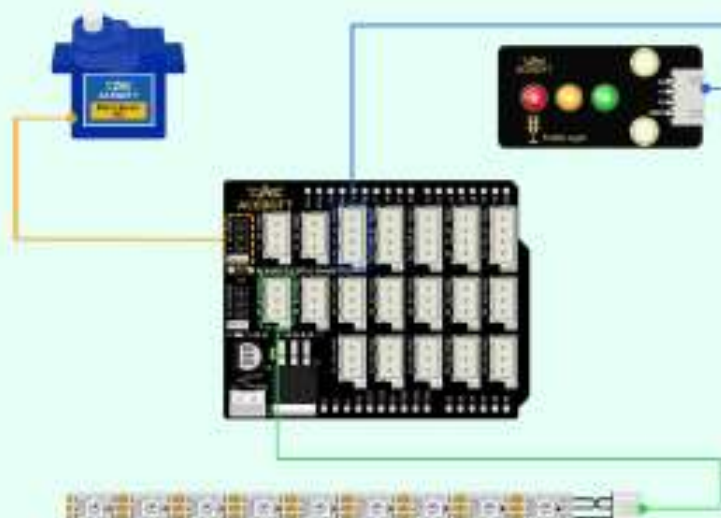
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	RGB Light Strip	1
	Traffic Light	1
	Servo	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 08 "08 Dopravní světla na zemi" montážního dokumentu.

3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <Adafruit_NeoPixel.h> //Calling the Lightstrip Library
#include <ESP32_Servo.h> //Calling the servo program library
#define R 14 //Define the pin of the traffic light red light
#define Y 2 //Define the pin of the traffic light yellow light
#define G 0 //Define the pin for the green light of the traffic light
#define Serve 4 //Declare the pins of the steering gear
#define LED_PIN 18 //Declares the pin of the RGB light strip
#define LED_COUNT 9 //Declares the number of beads on an RGB light band
#define BRIGHTNESS 50 //Set the brightness of the lamp bead. The range is
0-255
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN);
//Create a ribbon object
Servo MyServo; //Generates an object for the steering gear
const int railing_close = 90; //Configure the servo rail closing Angle
const int railing_open = 0; //Configure the servo rail opening Angle

void setup() {
    strip.begin(); //Initializes the light strip
    strip.setBrightness(BRIGHTNESS); //Set the brightness of the lamp strip
    pinMode(R, OUTPUT); //Set the red LED pin to output mode
    pinMode(Y, OUTPUT); //Set the yellow LED pin to output mode
    pinMode(G, OUTPUT); //Set the green LED pin to output mode
    MyServo.attach(Serve, 500, 2500); //Initialize the servo
    MyServo.write(railing_close); //Set the initial position of the servo
    delay(500);
}

void loop() {
    Traffic_Light(0, 0, 1); //Green light on
    for (int i = 0; i < LED_COUNT; i++) {
        //Take turns controlling each light bead on the light strip to turn
        red
        strip.setPixelColor(i, strip.Color(255, 0, 0));
        //Set the working color of the lamp beads to red:
        strip.show(); //Update the status of the light beads on the light strip
        delay(30); //Delay of 30ms
    }
    MyServo.write(railing_open); //Railing open
```

```

delay(2000); //Delay 2 second
Traffic_Light(0, 1, 0); //Yellow light on
delay(1000); //Delay 1 seconds
Traffic_Light(1, 0, 0); //Red light on
MyServo.write(railing_close); //Railing closed
for (int i = 0; i < LED_COUNT; i++) {
//Turn on green light for each LED on the LED strip in sequence
    strip.setPixelColor(i, strip.Color(0, 255, 0));
//Set the color of the LED to green
    strip.show(); //Update the state of the LEDs on the strip.
    delay(30); //Delay for 30 milliseconds
}
delay(3000); //Delay 3 second
}
//Define the traffic light color control function
//r, y, g are the parameters that control the operation of each
traffic light. The values are 0 or 1, corresponding to LOW and
HIGH
void Traffic_Light(int r, int y, int g) {
    digitalWrite(R, r);
    digitalWrite(Y, y);
    digitalWrite(G, g);
}

```

3. Vysvětlení instrukcí

Adafruit_NeoPixel.h	Knihovna pro ovládání RGB LED pásu obsahuje různé funkce pro řízení RGB LED pásů a souvisejících dat.
Adafruit_NeoPixel strip(count, pin):	Vytvořte objekt LED pásu s názvem objektu, který lze přizpůsobit. count je počet LED diod na pásu a pin je číslo pinu, ke kterému je pás připojen.
strip.begin()	Inicializujte LED pás.

<code>strip.setBrightness()</code>	Nastavte jas LED pásu v rozsahu 0-255.
<code>strip.Color(R, G, B)</code>	Nastavte hodnotu barvy LED diody, přičemž každý parametr je v rozsahu 0-255.
<code>strip.setPixelColor(i,color)</code>	Nastavte barvu specifikované LED diody, kde <i>i</i> je číslo LED diody a <i>color</i> je hodnota jasné barvy, získaná pomocí <code>strip.Color()</code> .
<code>strip.show()</code>	Aktualizujte stav každé LED diody na LED pásu.
<code>for</code>	Příkaz <code>for</code> je smyčkový příkaz, který slouží k opakovanému vykonávání určitého úseku instrukce. Jeho syntaxe je následující: <pre>For (initial expression; loop condition; incremental expression){ Statement block; }</pre> Initial expression: Používá se k inicializaci proměnných smyčky, provádí se pouze jednou před začátkem smyčky. Loop condition: Podmínka smyčky je zkontrolována před každou iterací. Pokud je pravdivá, vykoná se blok příkazů. Pokud je nepravdivá, smyčka končí. Statement block: Úsek příkazu, který je vykonáván v smyčce, uzavřený v složených závorkách. Incremental expression: Používá se k zvýšení nebo snížení hodnoty proměnné smyčky, provádí se po každém příkazu v bloku smyčky.

Sekce 3: Upozornění na bezpečnost přechodu



Cíle kurzu



-  **Naučit se principy bzučáku**
-  **Ovládnout použití bzučáku**
-  **Navrhnout a postavit varovný systém pro bezpečnost přechodů**



Kromě pozemních dopravních světel pro upozornění chodců, jaké další funkce můžeme navrhnout pro zlepšení bezpečnosti chodců?

Použití bzučáku k vydávání zvukových upozornění je skvělým vylepšením pro inteligentní chodníky pro chodce. Když je aktivní zelené světlo na chodníku, bzučák může vydávat zvuk, aby chodcům připomněl, aby rychle přešli. To nejen zvyšuje povědomí chodců o bezpečnosti, ale také účinně snižuje výskyt dopravních nehod.



V této lekci použijeme bzučák k simulaci inteligentního chodníku pro chodce se zvukovým upozorněním, což má za cíl zvýšit bezpečnost chodníků.



Viš to?

Jak vzniká zvuk?

Zvuk vzniká vibrací objektů. Když objekt vibruje, vzduch, který s ním přichází do kontaktu, také vibruje, což vytváří zvukové vlny. Tyto zvukové vlny se šíří vzduchem a nakonec vstupují do našich uší, což nám umožňuje vnímat zvuk.



Objekty schopné produkovat zvuk jsou definovány jako zvukové zdroje, jako jsou různé hudební nástroje. Různé zvukové zdroje vytvářejí zvuky s různými frekvencemi a zabarveními díky svým unikátním vzorcům vibrací. Přesně tak! Strunné nástroje vytvářejí zvuk vibracemi strun, dechové nástroje generují zvuk vibracemi vzduchu ve svých trubkách a bicí nástroje vytvářejí zvuk úderem nebo klepáním na jejich povrchy. Každý typ nástroje vytváří jedinečné zvuky na základě své konstrukce a způsobu hry.

Bzučák



Bzučák je běžná elektronická součástka používaná k vytváření zvukových signálů. Jeho princip fungování je

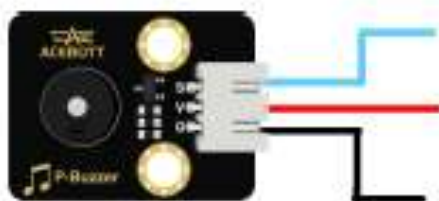


založen buď na piezoelektrickém efektu, nebo na principu elektromagnetické indukce. V piezoelektrickém bzučáku je piezoelektrická keramická součástka, která při aplikaci napětí na ni vibruje a vytváří zvuk. V elektromagnetickém bzučáku je elektromagnetická cívka a vibrující membrána. Když cívkou prochází proud, membrána vibruje, což rovněž vytváří zvuk.

Bzučáky lze rozdělit na aktivní a pasivní typy. Aktivní bzučák může přímo vydávat zvuk při připojení k napájecímu zdroji, zatímco pasivní bzučák vyžaduje externí oscilující signál, obvykle signál čtvercové vlny s proměnnou frekvencí, k tomu, aby vytvořil zvuk.

V tomto kurzu si ukážeme, jak používat pasivní bzučákový modul. Tento modul má tu vlastnost, že když jeho signálový pin přijme čtvercovou vlnu s určitou frekvencí, vydává odpovídající zvuk. Na platformě Arduino obvykle používáme funkci `tone()`, která generuje čtvercovou vlnu, aby ovládala pasivní bzučák a vytvářela zvuk.

Bzučák - Analýza pinů



S: Připojte I/O pin řídicí desky.

V: Připojte 5V pin řídicí desky.

G: Připojte GND pin řídicí desky.



Laboratoř ITS



Bezpečnost chodníku



Popis projektu



Na chodník je přidán bzučák na základě pozemního dopravního světla. Když je místní dopravní světlo zelené, bzučák vydává zvuk, aby chodce upozornil, aby rychle přešli chodník.



Hardwarové schéma

1. Seznam hardware

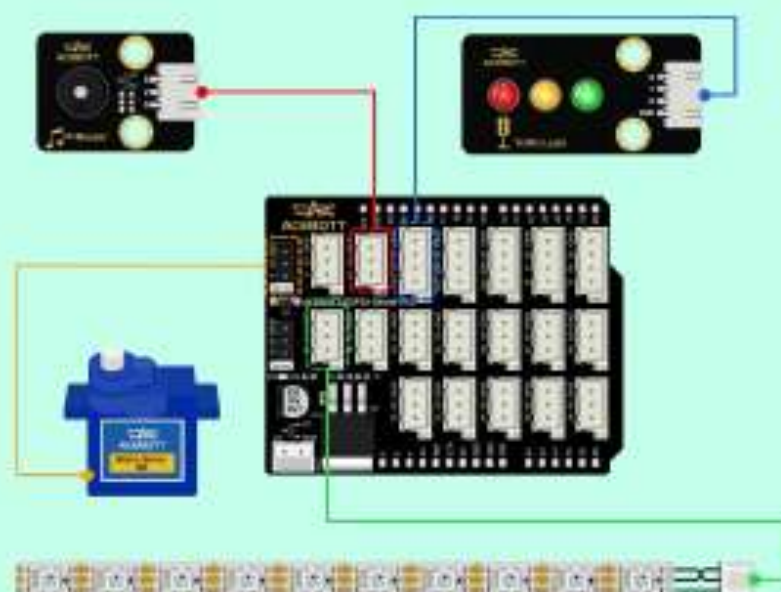
Obrázek	Název	Množství
	ESP32	1
	ESP32 expansion board	1
	RGB Light Strip	1
	Traffic Light	1
	Servo	1
	Buzzer	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 09 "09 Bezpečnostní připomenutí pro přechod" montážního dokumentu.

3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <Adafruit_NeoPixel.h> //Import the light strip library
#include <ESP32_Servo.h> //Calling the servo program library
#define R 14 //Define pins for red traffic lights
#define Y 2 //Define pins for yellow traffic lights
#define G 0 //Define pins for traffic lights and green lights
#define Serve 4 //Declare the pins of the servo motor
#define LED_PIN 18 //Declare the pins of the RGB light strip
#define LED_COUNT 9 //Declare the number of LED beads on the RGB strip
#define BRIGHTNESS 50 //Set the brightness of the lamp beads, ranging
from 0 to 255
#define buzzer 12 //Define the pins of the buzzer
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN); //Create a light
strip object
Servo MyServo; //Generate an object for a servo motor

void setup() {
    strip.begin(); //Initialize the light strip
    strip.setBrightness(BRIGHTNESS); //Set the brightness of the
light strip
    pinMode(R, OUTPUT); //Set the red light pin to output mode
    pinMode(Y, OUTPUT); //Set the yellow light pin to output mode
    pinMode(G, OUTPUT); //Set the green light pin to output mode
    pinMode(buzzer, OUTPUT); //Set the buzzer pin to output mode
    MyServo.attach(Serve, 500, 2500); //Initialize servo
    MyServo.write(90); //Configure servo initial position
    delay(500);
}

void loop() {
    Traffic_Light(0, 0, 1); //Green light on
    railing_up(); //Railing open
    for (int i = 0; i < LED_COUNT; i++) { //Take turns controlling each
light bead on the light strip to turn red
        strip.setPixelColor(i, strip.Color(255, 0, 0)); //Set the
working color of the lamp beads to red:
        strip.show(); //Update the status of the light beads on the light
strip
        delay(30); //Delay of 30ms
    }
    delay(2000); //Delay 2 seconds
    Traffic_Light(0, 1, 0); //Yellow light on
    delay(1000); //Delay 1 second
```

```

Traffic_Light(1, 0, 0); //Red light on
railing_down(); //Railing closed
for (int i = 0; i < LED_COUNT; i++) {
//Take turns controlling each light bead on the light strip to light
up the green light
    strip.setPixelColor(i, strip.Color(0, 255, 0));
//Set the working color of the lamp beads to green:
    strip.show();
//Update the status of the light beads on the light strip
    delay(30); //delay 30ms
}
for(int i = 0; i < 3;i++){
    tone(buzzer, 255); //Buzzer sounds
    delay(500); //Delay 500ms
    noTone(buzzer); //Buzzer off
    delay(500); //Delay 500ms
}
}
//Define traffic light color control function
void Traffic_Light(int r, int y, int g) {
    digitalWrite(R, r);
    digitalWrite(Y, y);
    digitalWrite(G, g);
}
void railing_up(){ //Define the opening angle of the railing
    MyServo.attach(Serve);
    for(int angle = 90; angle >= 0; angle = angle - 5){
        MyServo.write(angle);
    }
}
void railing_down(){ //Define railing closing angle
    MyServo.attach(Serve);
    for(int angle = 0; angle <= 90; angle = angle + 5){
        MyServo.write(angle);
    }
}
}

```

3. Vysvětlení instrukcí

<code>tone(pin,frequency,duration)</code>	Ovládejte specifikovaný pin, aby generoval čtvercovou vlnu s určenou frekvencí a řídil dobu jejího výstupu. 'pin' odkazuje na číslo specifikovaného pinu, 'frequency' je frekvence čtvercové vlny a 'duration' je doba trvání čtvercové vlny. Pokud není parametr 'duration' specifikován, čtvercová vlna bude pokračovat v generování.
<code>noTone(pin)</code>	Vypněte výstup čtvercové vlny na specifikovaném pinu.

Sekce 4: Tlačítkové dopravní světla



Cíle kurzu



- 📖 Pochopit funkci tlačítkových dopravních světel
- 📖 Naučit se princip fungování tlačítek
- 📖 Ovládnout metody řízení tlačítek
- 📖 Navrhnout a postavit dopravní světla pro chodce ve stylu tlačítka



Setkal(a) jsi se někdy se situacemi, kdy musíš čekat na červenou na chodníku příliš dlouho? Jak zlepšit tento jev?

V některých veřejných oblastech, jako jsou obchodní zóny, okolí škol a obytné čtvrti, kde je vysoký pohyb chodců a relativně nízký provoz vozidel, musí chodci často čekat dlouho na přechod silnice, což vede k přetížení chodců a potenciálním rizikům pro jejich bezpečnost. Tato situace je obzvláště



vážná během špičkových hodin, kdy dlouhá čekací doba nejen snižuje efektivitu dopravy, ale může také vést k dopravním nehodám.

Abychom tento problém vyřešili, můžeme implementovat tlačítková dopravní světla, která pomohou chodcům bezpečně přecházet silnici. Charakteristikou tlačítkových dopravních světel je, že když chodci stisknou tlačítko, světelná signalizace upraví své časování na základě aktuálních dopravních podmínek, což umožní chodcům pohodlnější přechod silnice. Tento přístup nabízí několik výhod:



1. Zvýšení pohodlí chodců: Tlačítková dopravní světla mohou upravit časování signálů v reálném čase podle poptávky chodců, což snižuje dobu čekání a zvyšuje efektivitu cestování.

2. Optimalizace rozdělení dopravních zdrojů: V oblastech s nižším vozidlovým provozem mohou tlačítková dopravní světla zabránit zbytečnému plýtvání zeleným světlem, čímž zlepšují využívání dopravních zdrojů.

3. Snížení dopravních rizik: Tlačítková dopravní světla mohou upravit časování signálů podle aktuálních podmínek na místě, což poskytuje chodcům více času na zeleném světle při přecházení silnice, a tím snižuje riziko dopravních nehod.

4. Úspora energie: V porovnání s tradičními dopravními světly mohou tlačítková dopravní světla upravit dobu svícení podle poptávky, čímž účinně snižují plýtvání energií.

5. Zlepšení městského řízení: Zavedení tlačítkových dopravních světel pomáhá městským správcům pochopit potřeby chodců při přecházení, čímž dále optimalizují časování dopravních signálů a zlepšují úroveň řízení městské dopravy.

Shrnuto, tlačítková dopravní světla, jako humanizované a inteligentní opatření pro řízení dopravy, mohou do jisté míry zmírnit přetížení chodců, zvýšit pohodlí chodců a zlepšit bezpečnost při přecházení silnice.

Tlačítková dopravní světla



Tlačítková dopravní světla fungují následovně: Když chodci potřebují přejít silnici, stisknou tlačítko, které vysílá signál do řadiče dopravního světla. Po obdržení signálu řadič upraví barvu světla pro chodce na základě stavu přepínání a logiky řízení světel, čímž vozidlům na silnici signalizuje, aby zpomalila a zastavila se, což poskytuje chodcům nezbytné dopravní vedení pro bezpečný přechod. Klíčovou součástí systému tlačítkových dopravních světel je funkce tlačítka. Nyní použijeme modul klávesnice k implementaci této tlačítkové funkce.



Tlačítko



Tlačítko, také známé jako stiskací tlačítko nebo přepínač, je mechanické zařízení, které slouží k otevření nebo uzavření elektrického obvodu. Obvykle se skládá z komponentů, jako jsou tlačítková kapka, pružina a kontakty. Stisknutím tlačítkové kapky se stlačí pružina, což způsobí kontakt nebo přerušení kontaktu s příslušným obvodem. Tlačítka jsou široce používána v různých elektronických zařízeních, jako jsou mobilní telefony, televizory a počítače.

V této lekci používáme modul tlačítkového obvodu, který se skládá z napájecího zdroje, tlačítka a souvisejících elektronických komponent. Tento obvod je běžně používán pro řízení vstupů a výstupů v elektronických obvodech. Když je tlačítko stisknuto, tlačítkový modul vydá signál nízké úrovně, a když je tlačítko uvolněno, vydá signál vysoké úrovně.

Tlačítko - Analýza pinů



S: Připojte I/O pin řídicí desky.

V: Připojte 5V pin řídicí desky.

G: Připojte GND pin řídicí desky.



Laboratoř ITS



Tlačítková tlačítka pro pěší semaforey



Popis projektu



Simulace funkce tlačítkového semaforu pro chodce: Stiskněte tlačítko, rozsvítí se červené světlo pro pruh a zelené světlo pro chodce, což signalizuje zákaz průjezdu vozidel a umožnění přechodu pro chodce.

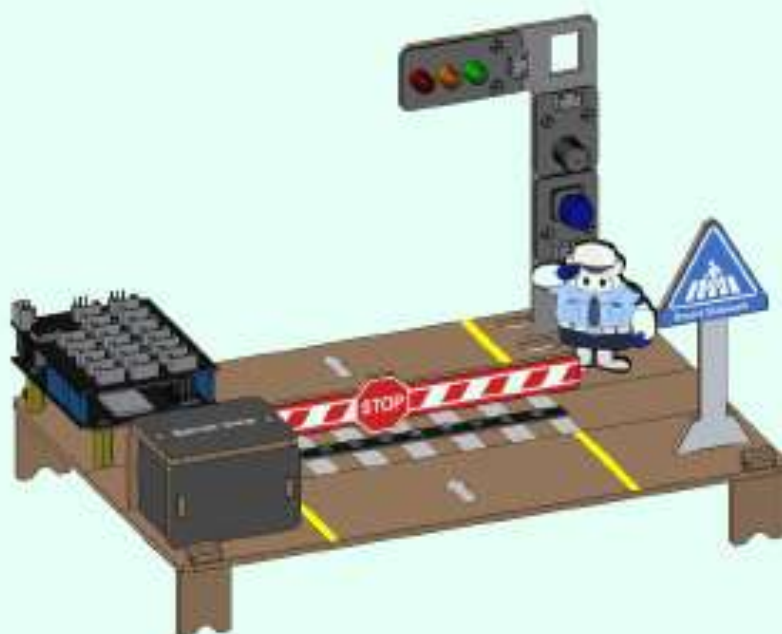


Hardwarové schéma

1. Seznam hardware

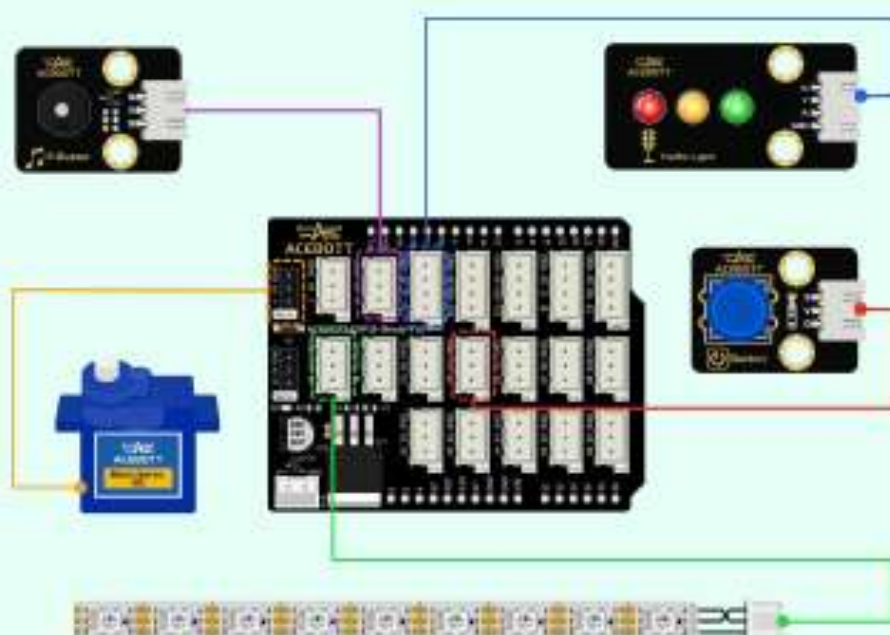
Obrázek	Název	Množství
	ESP32	1
	ESP32 Expansion Board	1
	RGB Light Strip	1
	Traffic Light	1
	Servo	1
	Button	1
	Buzzer	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 10 "10 Tlačítkové semaforey" montážního dokumentu.

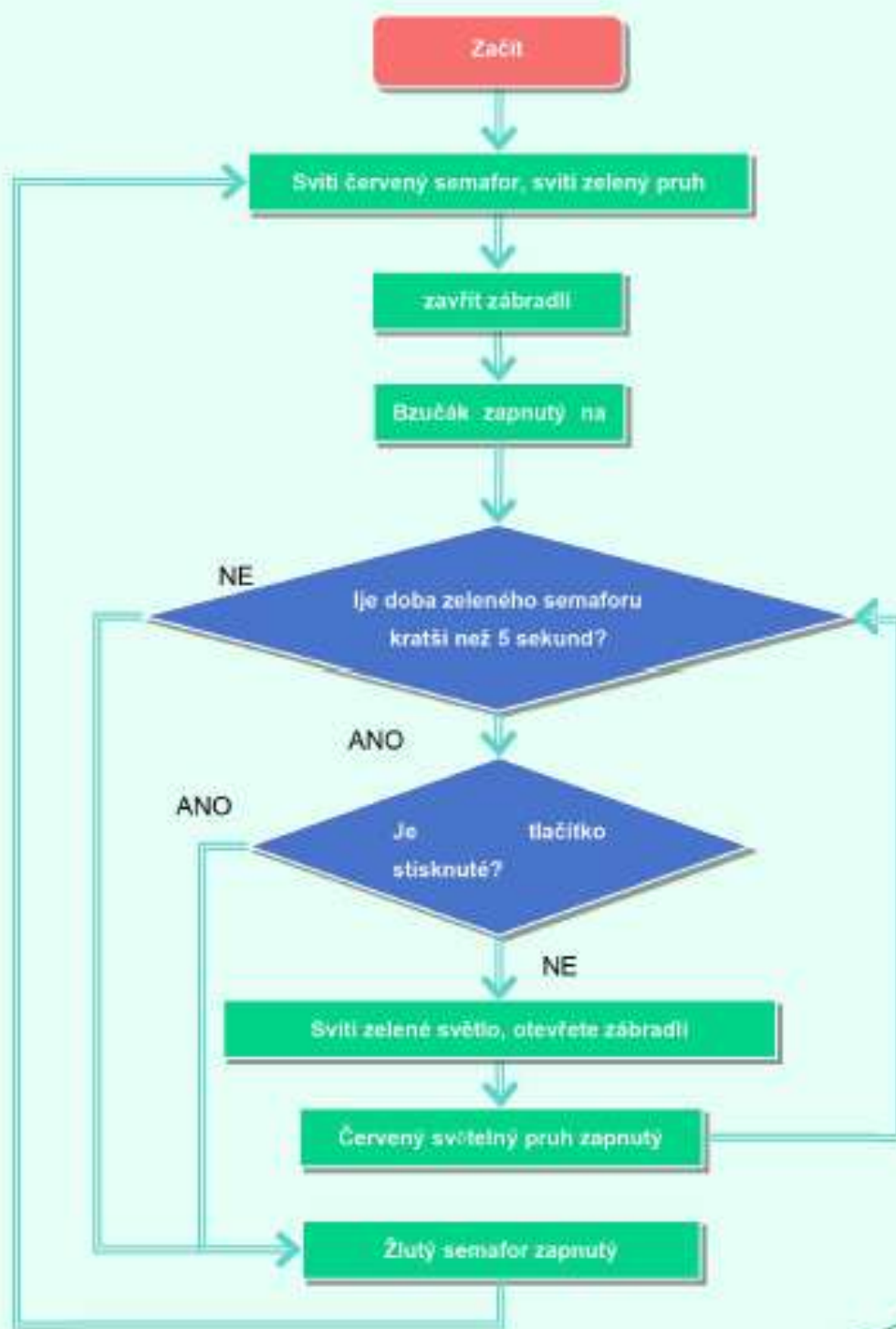
3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <Adafruit_NeoPixel.h> //Import the light strip library
#include <ESP32_Servo.h> //Import servo program library
#define R 14 //Define pins for red traffic lights
#define Y 2 //Define pins for yellow traffic lights
#define G 0 //Define pins for traffic lights and green lights
#define Serve 4 //Declare the pins of the servo motor
#define Button 25 //Declare the pins of the button
#define LED_PIN 18 //Declare the pins of the RGB light strip
#define LED_COUNT 9 //Declare the number of LED beads on the RGB strip
#define BRIGHTNESS 50 //Set the brightness of the lamp beads, ranging
from 0 to 255
#define buzzer 12 //Define the pins of the buzzer
int greentime = 5000; //Define the time for the green light on the
roadway
int lasttime; //Defining Timer Variables
Servo Myservo; //Configure the pins of the servo
int button_state;
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN); //Create a light strip
object

void setup() {
    strip.begin(); //Initializes the light strip
    strip.setBrightness(BRIGHTNESS); //Set the brightness of the light
strip
    pinMode(R, OUTPUT); //Set the red light pin to output mode
    pinMode(Y, OUTPUT); //Set the yellow light pin to output mode
    pinMode(G, OUTPUT); //Set the green pin to output mode
    pinMode(Button, INPUT); //Set key pins to input mode
    pinMode(buzzer, OUTPUT); //Set the buzzer pin to output mode
    Myservo.attach(Serve, 500, 2500); //Initialize servo
    Myservo.write(90); //Configure servo initial position
    delay(500);
}
```

```

void loop() {
    Traffic_Light(1, 0, 0); //Red light on
    railing_down(); //Railing closed
    for (int i = 0; i < LED_COUNT; i++) { //Take turns controlling each
light bead on the light strip to light up the green light
        strip.setPixelColor(i, strip.Color(0, 255, 0)); //Set the working
color of the lamp beads to green:
        strip.show(); //Update the status of the light beads on the light
strip
        delay(30); //Delay 30ms
    }
    for (int i = 0; i < 3; i++) {
        tone(buzzer, 255); //Buzzer sounds
        delay(500);
        noTone(buzzer); //Buzzer off
        delay(500);
    }
    lasttime = millis();
    while (millis() - lasttime < greentime) {
        if (digitalRead(Button) == 0) {
            delay(100);
            if (digitalRead(Button) == 0) {
                break; //Out of the loop
            }
        }
        Traffic_Light(0, 0, 1); //Green light on
        railing_up(); //Railing open
        for (int i = 0; i < LED_COUNT; i++) { //Take turns controlling
each light bead on the light strip to turn red
            strip.setPixelColor(i, strip.Color(255, 0, 0)); //Set the
working color of the lamp beads to red:
            strip.show(); //Update the status of the light beads on the light
strip
            delay(30); //Delay 30ms
        }
    }
    Traffic_Light(0, 1, 0); //Yellow light on
    delay(1000); //Delay 1 second
}

```

```

}
//Define traffic light color control function
void Traffic_Light(int r, int y, int g) {
    digitalWrite(R, r);
    digitalWrite(Y, y);
    digitalWrite(G, g);
}

void railing_up(){ //Define the opening angle of the railing
    MyServo.attach(Serve);
    for(int angle = 90;angle>=0;angle = angle-5){
        MyServo.write(angle);
    }
}

void railing_down(){//Define railing closing angle
    MyServo.attach(Serve);
    for(int angle = 0;angle<=90;angle = angle+5){
        MyServo.write(angle);
    }
}
}

```

Kapitola 5 Chytré parkoviště

V dnešním rychle se rozvíjícím městském prostředí a s rychlým nárůstem počtu automobilů se tradiční způsoby parkování ukázaly jako nedostatečné pro pokrytí potřeb lidí, což vede k rozšířeným problémům s parkováním. Vznik chytrých parkovišť efektivně zlepšil efektivitu a bezpečnost parkování, čímž poskytuje inovativní řešení pro městskou dopravu. V této kapitole se společně ponoříme do vědy a technologií za chytrými parkovišti a vydáme se na cestu plnou výzev a inovací.



Sekce 1 Inteligentní kontrola přístupu



Cíle kurzu



- 📖 Pochopit koncept inteligentních parkovišť
- 📖 Seznámit se s pojmem RFID
- 📖 Porozumět principům komunikace RFID
- 📖 Ovládnout používání RFID modulů
- 📖 Navrhnout a postavit inteligentní přístupový systém pro chytrá parkoviště



Už jste se někdy setkali s dopravní zácpou nebo obtížemi při hledání parkovacího místa, když jdete zaparkovat? Jak bychom to měli zlepšit?

V každodenním životě se často setkáváme s obtížemi při hledání parkovacích míst a dopravními zácpami, což nejenže plýtvá cenným časem, ale také zvyšuje nepohodlí při cestování. Pro řešení tohoto problému je zvláště důležitá výstavba chytrých parkovišť. Zavedením technologických řešení lze



výrazně zvýšit efektivitu parkování, zmírnit dopravní zácpy a poskytnout občanům pohodlnější a komfortnější zážitek z cestování.

Chytré parkoviště



Chytré parkoviště je typ parkovacího zařízení, které využívá pokročilé technologie a automatizované systémy k řízení parkovacího procesu, s cílem zlepšit efektivitu parkování a šetřit prostorové zdroje. Díky digitálnímu monitorování, inteligentním senzorovým zařízením a automatizovaným navigačním systémům umožňuje chytré parkoviště bezobslužný vjezd a výjezd vozidel, inteligentní navigaci parkováním a správu parkovacích míst v reálném čase. Na chytrém parkovišti vozidla autonomně provádějí parkování, vyzvedávání a další operace prostřednictvím rezervačních, rozpoznávacích a řídicích systémů, což významně snižuje parkovací nehody a zácpy způsobené lidskými faktory, čímž se zvyšuje efektivita a bezpečnost parkování. Pro městské řízení dopravy a optimalizaci parkovacích zdrojů se bezobslužná parkoviště stanou jedním z klíčových směrů budoucí dopravy.

V tradičních parkovištích je vjezd a výjezd vozidel řízen ručně, což často vede

k nízké efektivitě a častým dopravním zácpám. V chytrých parkovištích je pohyb vozidel zcela monitorován a řízen senzory, což umožňuje inteligentní řízení přístupu vozidel a zvyšuje efektivitu pohybu vozidel. V této lekci budeme simulovat inteligentní přístupový systém pro chytré parkoviště na základě RFID modulů a fotoelektrických senzorů.



RFID, což je zkratka pro Radio-Frequency Identification (identifikace pomocí rádiových vln), je automatická identifikační technologie, která využívá bezdrátové rádiové signály k identifikaci cílů a získávání relevantních dat. Její základní princip spočívá ve využívání elektromagnetických polí a rádiové frekvenční komunikace k bezdrátovému přenosu informací uložených v RFID značkách do RFID čteček, což umožňuje sledování, identifikaci a správu cílových objektů. Technologie RFID nachází široké uplatnění v oblastech, jako je řízení logistiky, sledování zásob, správa majetku, inteligentní doprava a rozpoznávání identity.

RFID systém se obvykle skládá ze tří hlavních komponent: RFID značek, RFID čteček/zapisovačů a backendového systému.

RFID značky jsou elektronické štítky vybavené čipy a anténami, které umožňují rozpoznávání objektů a ukládání dat. Když se RFID značka přiblíží k RFID čtečce, čtečka



generuje signál rádiové frekvence určité frekvence a výkonu, který aktivuje RFID značku. Po aktivaci RFID značka přijme energii RF signálu a použije svůj pracovní obvod k demodulaci a zpracování dat, které následně vrátí uložené údaje v čipu zpět čtečce/zapisovači prostřednictvím RF signálu.

RFID čtečka/zapisovač je zařízení sloužící k čtení a zápisu dat z RFID značek. Může vysílat rádiové frekvenční signály, přijímat data vrácená RFID značkami a komunikovat s RFID značkami pomocí rádiové frekvenční komunikace. Čtečky



obvykle zahrnují komponenty jako antény, RF moduly, procesory atd., které umožňují rozpoznání RFID značek a čtení a zápis dat.

Backendový systém je zodpovědný za příjem, zpracování a ukládání dat získaných z RFID systému. Připojením k RFID čtečkám a zapisovačům může backendový systém v reálném čase získávat informace o značkách, provádět analýzu dat, zaznamenávat sledování a umožňovat sledování a správu objektů.

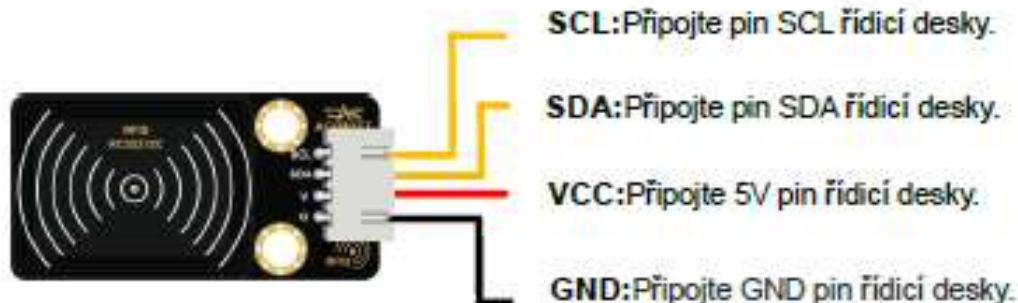


RFID RC255 Modul



RFID modul RC522 IIC je běžný modul pro identifikaci pomocí rádiové frekvence (RFID), který umožňuje bezdrátovou identifikaci a čtení/zápis dat z blízkých elektronických štítků. Komunikuje s hostitelským řadičem nebo mikrokontrolérem prostřednictvím rozhraní IIC.

RFID - Analýza pinů



IC karta



IC karta, známá také jako karta s integrovaným obvodem, má schopnost zapisovat a ukládat data. IC karty existují ve dvou formách: kontaktní a bezkontaktní. IC karta použitá v této lekci je bezkontaktní a komunikuje s čtečkou/zapisovačem prostřednictvím RFID technologie.





Laboratoř ITS



Intelligentní řízení přístupu



Popis projektu



Použijte RFID a infračervené senzory pro vyhýbání se překážkám k ovládání otevírání a zavírání zábrany parkoviště. Když infračervený senzor detekuje vozidlo, zábrana parkoviště se otevře, což umožní vozidlu vjet na parkoviště. Když RFID modul detekuje IC kartu, zábrana parkoviště se otevře, což umožní vozidlu opustit parkoviště.



Hardwarové schéma

1. Seznam hardware

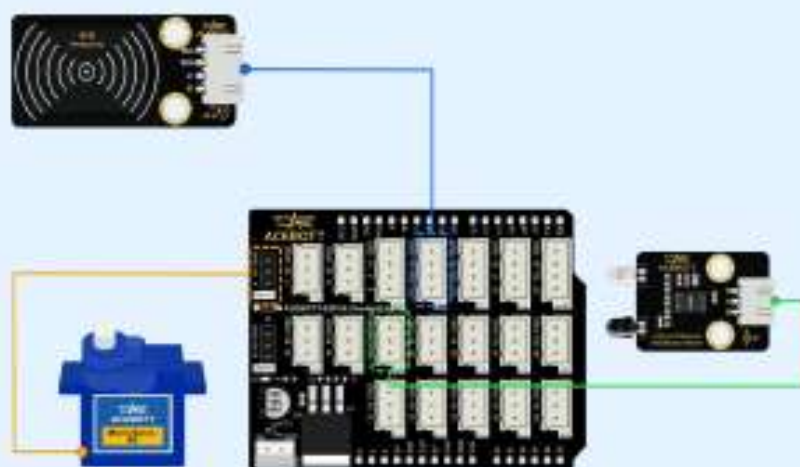
Obrázek	Název	Množství
	ESP32 Board	1
	ESP32 Expansion Board	1
	Servo	1
	RFID module	1
	Infrared obstacle avoidance sensor	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 11 "11 Inteligentní přístupová kontrola" montážního dokumentu.

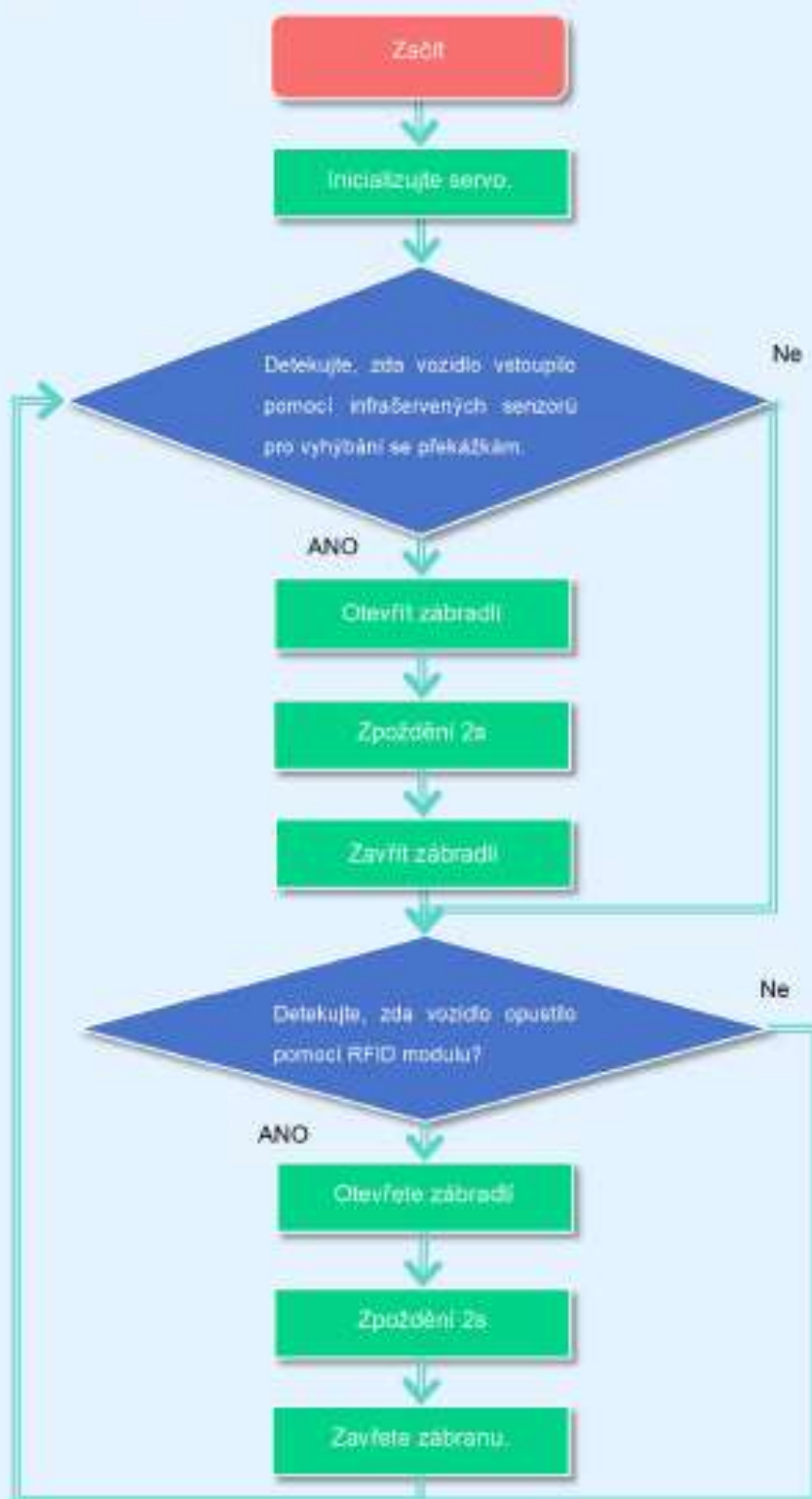
3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <ESP32_Servo.h> //Calling the servo program library
#include <MFRC522_I2C.h> //Call RFID library
#define serve 4 //Define the pins of the servo sensor
#define PE 23 //Define the pins of the infrared obstacle
avoidance sensor
int infrared_Sensor ; //Define variable values for infrared
obstacle avoidance sensors
MFRC522_I2C mfrc522(0x28, -1); //Configure RFID address
String rfid_str = ""; //Define variables of string type to read
card numbers

Servo Myservo; //Generate an object for a servo motor
const int railing_close = 90; //Configure the angle of the servo
railing
const int railing_open = 0; //Configure servo railing opening
angle

void setup() {
    Serial.begin(9600); //Initialize baud rate
    Wire.begin(); //Initialize i2c serial port
    mfrc522.PCD_Init(); //Initialize RFID module
    pinMode(PE, INPUT); //Set the pins of the infrared obstacle
avoidance sensor to input mode
    Myservo.attach(serve, 500, 2500); //Initialize servo
    Myservo.write(railing_close); //Configure servo initial
position
    delay(500);
}
void loop() {
    // Check if any vehicles have passed through
    if (digitalRead(PE) == 0) {
        Myservo.write(railing_open); //Open the railing
        delay(2000);
        Myservo.write(railing_close); //Close the railing
    }
}
```

```

//Check if any vehicles have left
if(!mfr522.PICC_IsNewCardPresent() || !mfr522.PICC_ReadCardSerial()) {
    delay(50);
    return; //Return to the main program and restart
}
else{
    Myservo.write(railing_open); //Open the railing
    delay(2000);
    Myservo.write(railing_close); //Close the railing
}
}

```

3. Vysvětlení instrukcí

MFRC522_I2C.h	RFID knihovna
mfr522.PICC_IsNewCardPresent()	Zkontrolujte, zda je v blízkosti čtečky/zapisovače RFID nová IC karta.
mfr522.PICC_ReadCardSerial()	Přečtěte sériové číslo IC karty.

Sekce 2 Zobrazení parkovacího místa



Cíle kurzu



- 🔧 Prozkoumat správu parkovacích míst v parkovištích
- 🔧 Naučit se princip obrazovky LCD1602
- 🔧 Ovládnout používání obrazovky LCD1602
- 🔧 Navrhnout a vytvořit systém zobrazení parkovacích míst pro inteligentní parkoviště



Už jste někdy zažili situace, kdy stále hledáte volné parkovací místo na parkovišti?

Problém s hledáním parkovacích míst na parkovištích je běžnou situací, se kterou se mnoho lidí setkává v každodenním životě. Ať už při nakupování v obchodních centrech, práci v kancelářských budovách nebo návštěvě veřejných míst, nerovnováha mezi nabídkou a poptávkou po parkovacích místech často nutí řidiče trávit značné množství času hledáním vhodného místa. To



nejen plýtvá časem a energií jednotlivců, ale také představuje výzvy pro okolní dopravní provoz a správu parkování. Proto je zvláště důležité racionální plánování a správa parkovacích zdrojů. To může účinně snížit plýtvání parkovacími místy, zlepšit efektivitu parkování a zvýšit celkový zážitek z parkování.

Existuje několik důvodů, proč mohou být parkovací místa nedostupná:

Dlouhodobé parkování: Některá vozidla mohou zůstat zaparkována na místech po delší dobu, aniž by se přesunula, čímž brání ostatním řidičům v jejich využití. Dlouhodobé parkování může nastat v důsledku toho, že majitelé nechávají svá vozidla bez dozoru na delší dobu, nebo kvůli nedostatečné správě parkovacího zařízení.



Špatně naplánované parkovací plochy: Parkovací místa mohou být nedostatečně naplánovaná, nerovnoměrně rozložená nebo jich může být nedostatek, což vede k nerovnováze mezi nabídkou a poptávkou. Například některé oblasti mohou být hustě zaplněny parkovacími místy, zatímco jiné jich mají málo.

vizuální zážitky

Nedostatečné parkovací vybavení: Nejasné označení parkovacích míst, nemožnost zobrazit informace o obsazenosti a další faktory mohou přispět k nedostupnosti parkovacích míst.

V této lekci vytvoříme systém zobrazení parkovacích míst, který využívá LCD obrazovku k zobrazení informací o parkovacích místech. Tento systém pomůže lidem lépe porozumět dostupnosti parkovacích míst na parkovišti, čímžlepší efektivitu parkování.



LCD (displej z tekutých krystalů) je zařízení, které využívá technologii tekutých

krystalů k zobrazení obrazů a textu.

Displej se skládá z vrstev včetně vrstvy tekutých krystalů, polarizátorů, světelného vodiče a podsvícení.

Kontrolou orientace molekul tekutých krystalů lze regulovat průchod světla, a

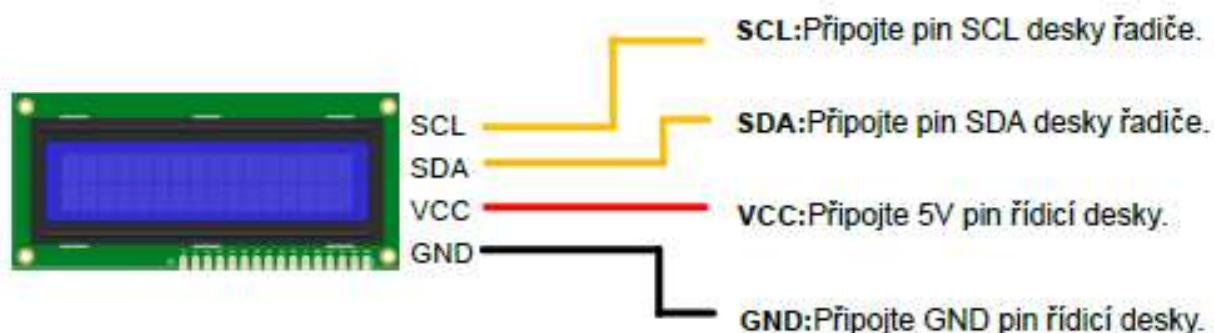
tím i zobrazování obrazu. S neustálým vývojem technologie LCD displeje zaznamenaly zlepšení v rozlišení, rychlosti odezvy, reprodukci barev a dalších aspektech, což splňuje nároky uživatelů na vysoce kvalitní zobrazení. V oblastech, jako jsou chytré telefony, monitory počítačů a televizory, se LCD displeje staly hlavní volbou a poskytují uživatelům realističtější a jasnější

LCD1602 Displejový modul



V této lekci budeme používat displej LCD1602, což je běžně používaný znakový LCD modul. Skládá se ze znakového LCD displeje, řídicího a hnacího hlavního obvodu, rozšiřujícího hnacího obvodu a několika rezistorů, kondenzátorů a konstrukčních součástí namontovaných na PCB desce. Výhodami tohoto displeje jsou nízká spotřeba energie, malá velikost a nízké vyzařování. Tento modul podporuje komunikaci pomocí protokolu I2C, což šetří zdroje pinů na řídicí desce. V tomto kurzu použijeme LCD1602 modul k zobrazení informací o obsazenosti parkoviště.

LCD1602 Displejový modul - Analýza pinů





Laboratoř ITS



Zobrazení parkovacích míst



Popis projektu



Pomocí infračervených senzorů pro vyhýbání se překážkám sledujeme příchozí vozidla, zatímco modul RFID sleduje odjíždějící vozidla a na displeji zobrazuje vypočítaný počet zbývajících a celkových parkovacích míst.

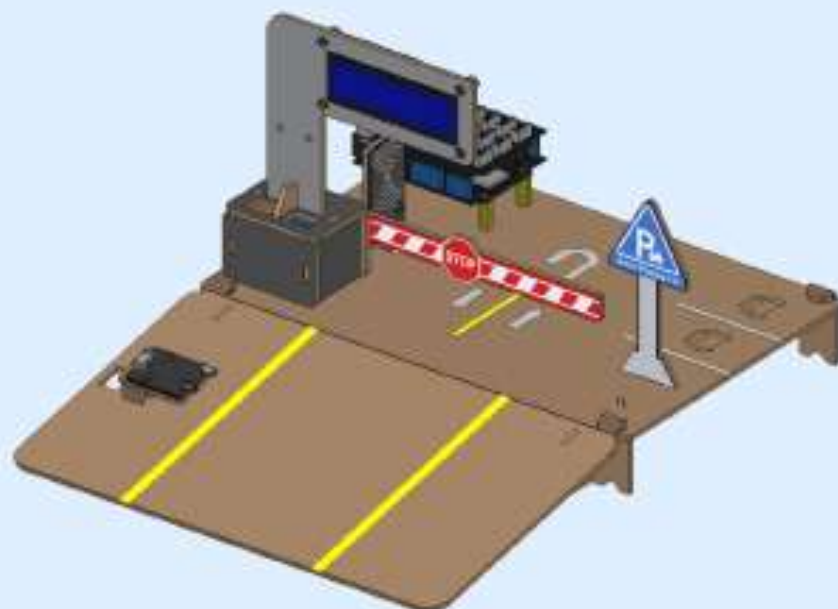


Hardwarové schéma

1. Seznam hardware

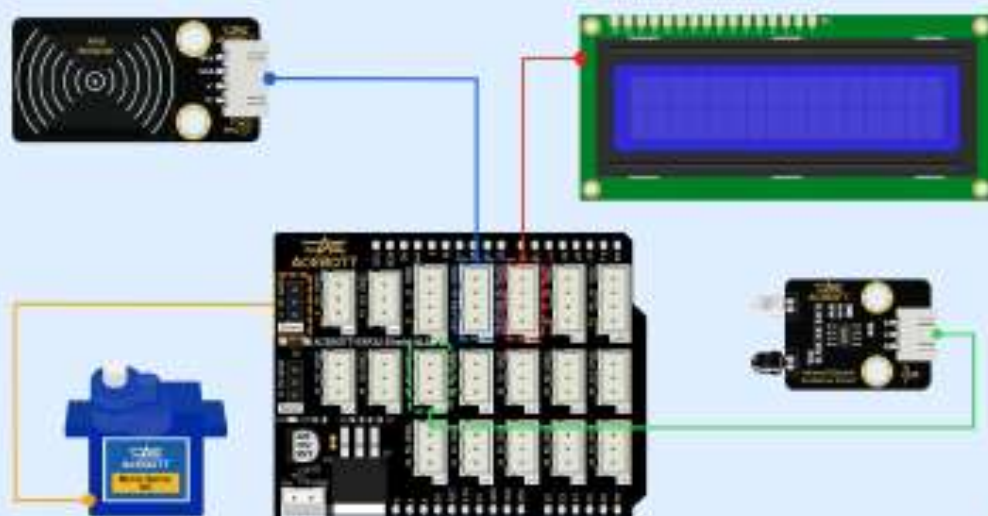
Obrázek	Název	Množství
	ESP32 Board	1
	ESP32 Expansion Board	1
	LCD1602	1
	Infrared Obstacle Avoidance Sensor	1
	Servo	1
	RFID Module	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 12 "12: Zobrazení parkovacích míst" montážního dokumentu.

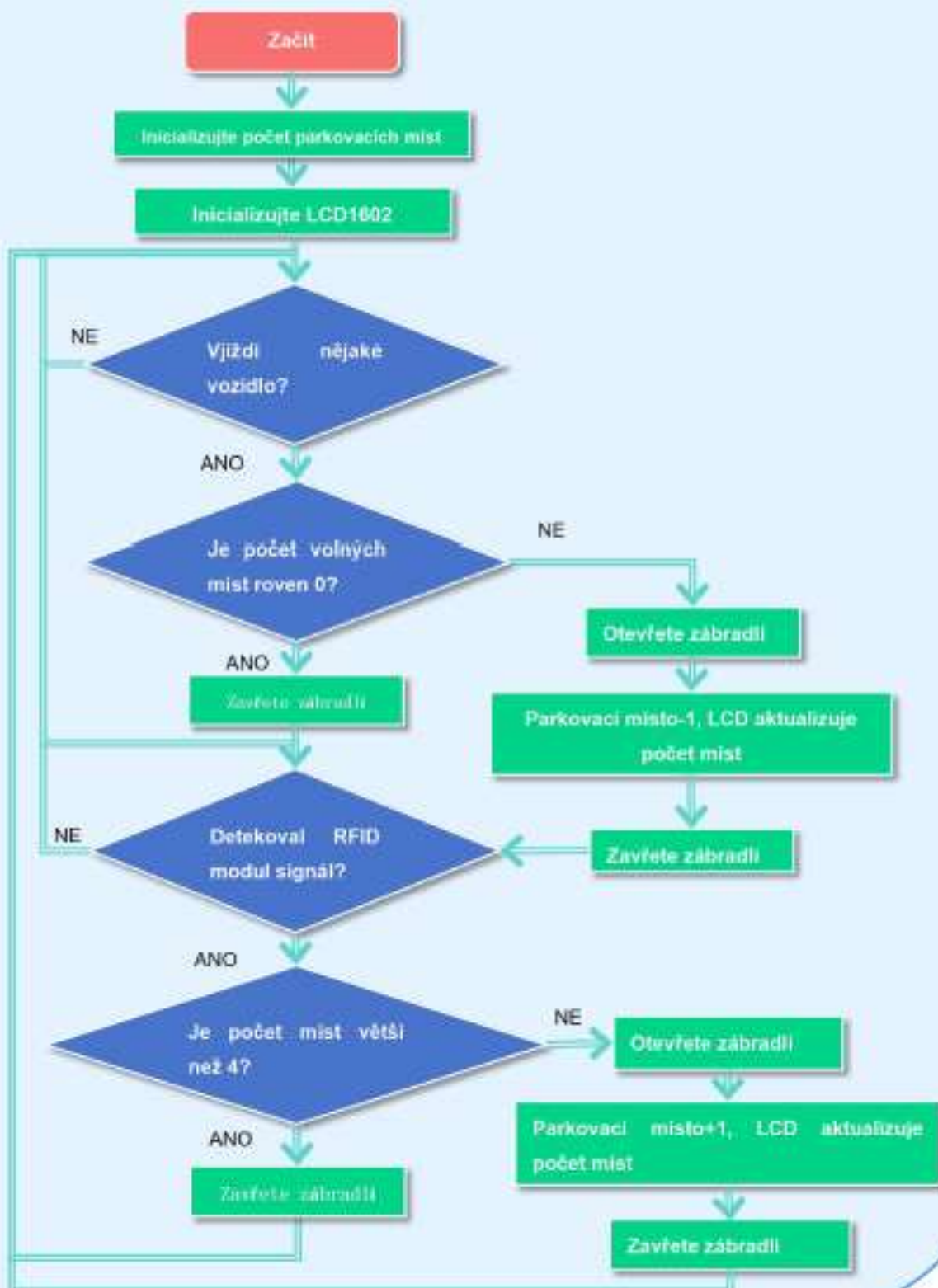
3. Schéma zapojení hardwaru





Programování

1. Programový blokový diagram



2. Kódování

```
#include <Wire.h> //Import the library of i2c
#include <MFRC522_I2C.h> //Import RFID library
#include <ESP32_Servo.h> //Import rudder hangar
#include <hd44780.h> //Import LCD Library
#include <hd44780ioClass/hd44780_I2Cexp.h> //Import LCD
Library
#define serve 4 //Define the pins of the servo sensor
#define PE 23 //Define the pins of the infrared obstacle avoidance
sensor
int infrared_Sensor; //Define variables for infrared obstacle
avoidance sensors
MFRC522_I2C mfrc522(0x28, -1); //Configure RFID address
String rfid_str = ""; //Define variables of string type to read
card numbers
Servo Myservo; //Generate an object for a servo motor
const int railing_close = 90; //Define the angle at which the
railing is closed
const int railing_open = 0; //Define the angle at which the
railing opens
const int i2cAddress = 0x27; //I2C address of LCD1602
const int numRows = 2; //Number of rows in LCD1602
const int numCols = 16; //Number of columns in LCD1602
hd44780_I2Cexp lcd(i2cAddress, numRows, numCols); //Create
LCD1602 object
int Parking_space = 5; //Define the number of parking spaces
void ledspace() { //Display real-time parking spaces
    lcd.setCursor(0, 1);
    lcd.print("Space:");
    lcd.setCursor(6, 1);
    lcd.print(Parking_space);
    delay(200);
}

void setup() {
    Serial.begin(9600); //Set the baud rate to 9600
    delay(100);
    Wire.begin(); //Initialize i2c serial port
    mfrc522.PCD_Init(); //Initialize RFID module
    Myservo.attach(serve, 500, 2500); //Initialize servo
    lcd.begin(numCols, numRows); //Initialize LCD1602
```

```

    lcd.backlight();
    pinMode(PE, INPUT); //Set the pins of the infrared obstacle
avoidance sensor to input mode
    delay(500);
    lcd.clear(); //Display real-time parking space quantity
}

void loop() {
//Display the total number of parking spaces
    lcd.setCursor(0, 0);
    lcd.print("TtalNumber:");
    lcd.setCursor(11, 0);
    lcd.print(5);
    ledspace(); //Display real-time number of parking spaces
    infrared_Sensor= digitalRead(PE); //Obtain the values of
infrared obstacle avoidance sensors
    Serial.println(infrared_Sensor);
// Check for vehicle entry
    if (digitalRead(PE)== 0) {
        if(Parking_space == 0){
            Myservo.write(railing_close); //Close the railing
        }
        else{
            Myservo.write(railing_open); //Open the railing
            delay(2000);
            Myservo.write(railing_close); //Close the railing
            Parking_space -= 1; //Number of parking spaces -1
            ledspace(); //Display real-time parking space quantity
        }
    }
// Check if any vehicles have left
    if(!mfrc522.PICC_IsNewCardPresent()|| !mfrc522.PICC_ReadCardSerial(
)) {
        delay(50);
        return; //Return to the main program and restart
    } else {
        if (Parking_space > 4) {
            Myservo.write(railing_close); //Close the railing
        } else {
            Myservo.write(railing_open); //Open the railing
            delay(2000);
            Myservo.write(railing_close); //Close the railing
            Parking_space += 1; //Number of parking spaces +1
        }
    }
}

```

```

    ledspace();//Display real-time parking space quantity
  }
}
}

```

3. Vysvětlení instrukcí

<code>hd44780.h</code>	Knihovna pro LCD displej.
<code>hd44780_I2Cexp lcd ()</code>	Vytvořte objekt LCD1602.
<code>lcd.setCursor(columnnumber,rownumber)</code>	Tato funkce umístí kurzor LCD na požadovanou pozici. Veškerý text zobrazený po této funkci začne od uvedené pozice. Například použijte: <code>lcd.setCursor(4, 0)</code>, což je pátý sloupec a první řádek (protože počítání začíná od 0,0).
<code>lcd.print("text")</code>	Tato funkce se používá k tisku textu na LCD. Bez ohledu na to, jaký řetězec je vložen, bude zobrazen na LCD.

Sekce 3 Hlasový vysílací systém



Cíle kurzu



- 🚗 Pochopit principy MP3 přehrávačů
- 🚗 Ovládnout používání MP3 přehrávačů
- 🚗 Navrhnout a vytvořit inteligentní parkovací systém s hlasovým vysíláním



Kromě základních funkcí parkovišť, jaké další funkce můžeme navrhnout, aby byla parkoviště inteligentnější?

Kromě funkcí, jako je zobrazení volných parkovacích míst, inteligentní přístupový systém a systémy prodeje lístků, mohou chytrá parkoviště také zahrnovat pokročilé hlasové vysílání. Prostřednictvím hlasového vysílání mohou chytrá parkoviště plnit různé praktické funkce: mohou poskytovat řidičům



signály

informace o způsobech platby za parkování, částkách poplatků a další související informace, čímž zvyšují efektivitu a pohodlí platebních procesů. Funkce hlasového vysílání může být také využita k vysílání důležitých oznámení, jako jsou varování o počasí, nouzová upozornění apod., což zlepšuje celkovou efektivitu řízení.

Jak přidat funkci hlasového vysílání do chytrých parkovišť?



Přidání funkce hlasového vysílání vyžaduje použití hlasového přehrávače, a existuje mnoho typů hlasových přehrávačů, včetně MP3, MP4 a dalších. Tyto přehrávače mají všechny vysokou schopnost zpracování zvuku v HD

kvalitě. V této části se zaměříme na představení MP3 přehrávače a podrobně se podíváme na to, jak používat modul pro přehrávání MP3 k přehrávání hlasu.

Co je MP3?



MP3 přehrávač je digitální audio zařízení navržené speciálně pro přehrávání hudby ve formátu MP3. Obvykle se skládá z mikroprocesoru, paměti a audio dekodéru. Mikroprocesor řídí činnost zařízení, paměť ukládá audio



soubory a audio dekodér dekóduje audio soubory na slyšitelné zvukové

MP3 Modul



V této lekci používáme modul pro přehrávání MP3, který zahrnuje nezávislý procesor, TF paměťovou kartu a reproduktor. Pomocí sériové komunikace můžeme ovládat modul pro přehrávání MP3 a přehrávat hudbu z ESP32. MP3 modul má čtyři piny: RX, pro odesílání dat na řídicí desku; TX, pro přijímání dat z řídicí desky; G a V, pro napájení modulu MP3 přehrávače.

MP3 - Analýza Pinů



RX:Připojte I/O pin řídicí desky.

TX:Připojte I/O pin řídicí desky.

V:Připojte 5V pin řídicí desky.

G:Připojte GND pin řídicí desky.



Tipy

Před ovládáním modulu pro přehrávání MP3, aby přehrával hudbu, je nutné předem uložit písně na paměťovou kartu MP3 modulu. Níže jsou kroky pro uložení písní na MP3 přehrávač:

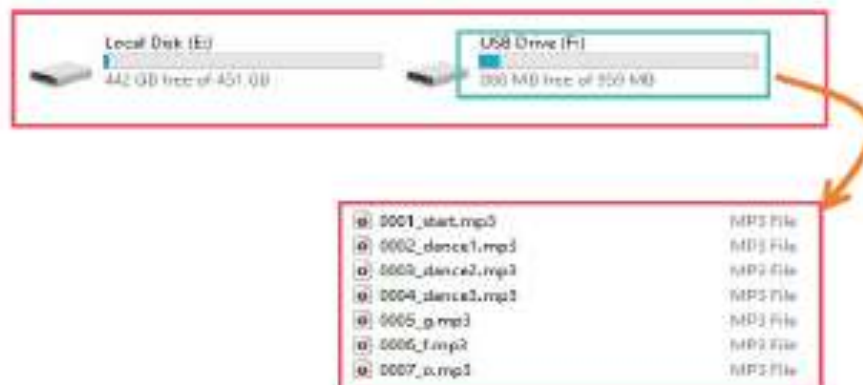
(1) Vložte paměťovou kartu do slotu pro kartu na MP3 modulu.



(2) Na zadní straně modulu pro přehrávání MP3 se nachází USB rozhraní. Připojte USB rozhraní MP3 modulu k počítači pomocí datového kabelu typu C.



(3) Po připojení MP3 modulu k počítači se objeví nová složka. Umístěte MP3 soubory do této složky. Pokud jsou soubory v kořenovém adresáři, jejich název musí začínat čtyřmi číslicemi, např. '0001Hero.mp3', '0002Jingle bells.mp3'.



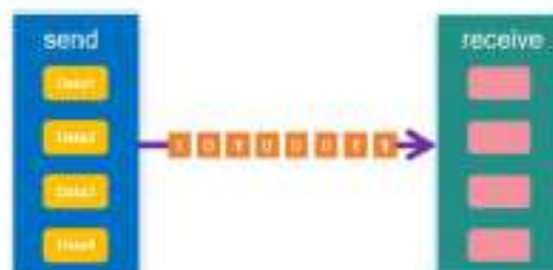
(4) Pokud chcete v kořenovém adresáři karty vytvořit složku, pojmenujte ji dvěma číslicemi a MP3 soubory uvnitř složky třemi číslicemi.



Sériová komunikace



Sériová komunikace (UART) je typ protokolu sériového přenosu dat, který se používá k výměně dat mezi dvěma zařízeními. Při sériové komunikaci se data přenášejí po



jednotlivých bitech v sekvenci a při předem stanovené přenosové rychlosti.

Sériová komunikace se často používá pro výměnu dat mezi vestavěnými systémy, mikrokontroléry, senzory, akčními členy a dalšími zařízeními. Například komunikace mezi řídicí deskou a počítačem nebo mezi senzorem a vestavěným zařízením může být realizována prostřednictvím sériových portů. Tento způsob komunikace je jednoduchý a stabilní, což ho činí vhodným pro mnoho aplikačních scénářů.

Sériová komunikace vysílá a přijímá data přes dva porty: TX a RX. TX je port pro odesílání dat, zatímco RX je port pro přijímání dat. V ESP32 jsou tři kanály sériové komunikace: UART0, UART1 a UART2, z nichž každý má přiřazené výchozí I/O piny.

UART0		UART1		UART2	
TX	RX	TX	RX	TX	RX
1	3	10	9	17	16



Tipy

V ESP32 se výchozí UART0 používá pro stahování programu a ladění. Pokud chcete použít UART0 ke komunikaci s externími zařízeními, je nutné před nahráním nebo laděním programu odpojit externí zařízení na UART0. Kromě toho nejsou piny pro tři kanály sériové komunikace na desce ESP32 pevně přiřazeny; mohou být namapovány na libovolné piny podle vašich potřeb.



Laboratoř ITS



Hlasový vysílací systém



Popis projektu



V inteligentním parkovišti přidejte MP3 modul pro přehrávání hlasových zpráv, jako například „Vítejte“ a „Šťastnou cestu“.

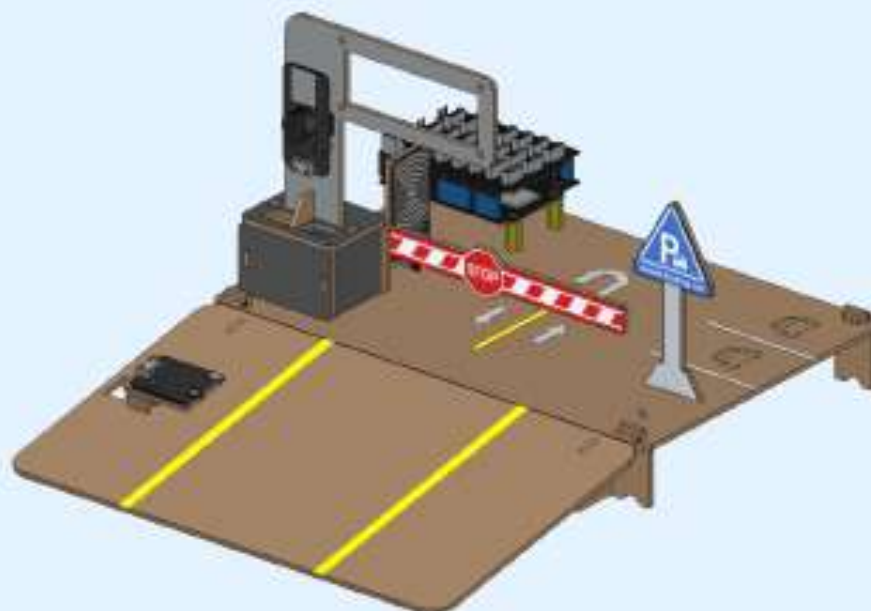


Hardwarové schéma

1. Seznam hardware

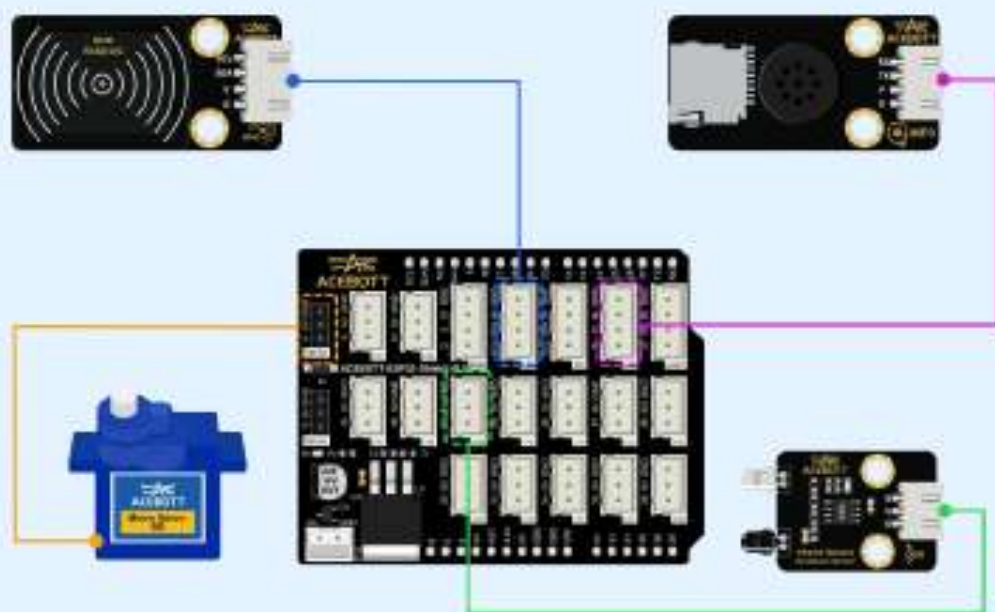
Obrázek	Název	Množství
	ESP32 Board	1
	ESP32 Expansion Board	1
	Servo	1
	RFIDModule	1
	MP3 Module	1
	Infrared Obstacle Avoidance Sensor	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 13 "13 Hlasový systém vysílání" montážního dokumentu.

3. Schéma zapojení hardwaru



2. Kódování

```
#include <ACB_MP3.h>      //Import MP3 library
#include <MFRC522_I2C.h>  //Import RFID library
#include <ESP32_Servo.h>  //Import servo library
#include <Wire.h> //Import the library of i2c
#define serve 4 //Define the pins of the servo sensor
#define PE 23 //Define the pins of the infrared obstacle avoidance
sensor
int infrared_Sensor;      //Define variable values for infrared
obstacle avoidance sensors
MFRC522_I2C mfrc522(0x28, -1); //Configure RFID address
String rfid_str = ""; //Define variables of string type to read
card numbers
Servo Myservo; //Generate an object for a servo motor
const int railing_close = 90; //Define the angle at which the
door servo closes the door
const int railing_open = 0; //Define the angle at which the door
servo opens the door
ACB_MP3 MyMP3; //Generate an MP3 object

void setup() {
    Wire.begin(); //Initialize i2c serial port
    pinMode(PE, INPUT); //Set the pins of the infrared obstacle
avoidance sensor to input mode
    mfrc522.PCD_Init(); //Initialize RFID module
    MyMP3.init(16,17); //Initialize the MP3 module
    Myservo.attach(serve, 500, 2500); //Initialize servo
    Myservo.write(railing_close); //Turn the servo mechanism to its
initial position
    delay(500); //Waiting for servo rotation to end
}

void loop() {
    infrared_Sensor= digitalRead(PE); //Read the value of the
infrared obstacle avoidance sensor
```

```

// Check for vehicle entry
if (digitalRead(PE)== 0) {
    Myservo.write(railing_open); //Open the railing
    MyMP3.playInFolder(0x02); //MP3 player playing "Welcome"
    delay(2000);
    Myservo.write(railing_close); //Close the railing
}

// Check if any vehicles have left
if(!mfrc522.PICC_IsNewCardPresent()||!mfrc522.PICC_ReadCard
Serial()){
    //If no new card exists or the card serial number is not
    successfully read
    delay(50);
    return; //Return to the main program and restart
} else {
    Myservo.write(railing_open); //Open the railing
    MyMP3.playInFolder(0x01); //MP3 player playing "Save journey"
    delay(2000);
    Myservo.write(railing_close); // Close the railing
}

```

3. Vysvětlení instrukcí

<code>ACB_MP3.h</code>	Hlavní soubor knihovny pro modul přehrávání MP3.
<code>ACB_MP3 MyMP3</code>	Vytvořte objekt pro MP3.
<code>MyMP3.playInFolder(p)</code>	Ovládejte MP3 přehrávač, aby přehrál skladby na specifikovaném místě ve složce na paměťové kartě, kde 'p' představuje číslo skladby k přehrání v hexadecimálním formátu.

Sekce 4 Integrovaný systém chytrého parkoviště



Cíle kurzu



🚧 Navrhování a výstavba integrovaného chytrého parkovacího systému



Jak funguje kompletní chytré parkoviště?



IV předchozích lekcích jsme se seznámili s funkcemi a principy různých aplikací v inteligentních parkovacích systémech. Jaký je však efekt jejich kombinace?

V této lekci navrhujeme a vybudujeme kompletní systém pro řízení chytrého parkování. Začneme přezkoumáním každého klíčového modulu – inteligentní přístupové kontroly, zobrazení parkovacích míst a inteligentního hlasového vysílání. Každý modul je nezbytnou součástí systému, která vyžaduje precizní integraci a hladkou koordinaci. Prostřednictvím dnešního studia nejen implementujeme technické aplikace, ale

také získáme hluboké porozumění tomu, jak tyto technologie řeší reálné problémy prostřednictvím praktického provozu.



Inteligentní přístupová kontrola parkoviště: Ovládání otevírání a zavírání závor na parkovišti pomocí RFID a fotoelektrických senzorů. Když fotoelektrický senzor detekuje vozidlo, závora se automaticky otevře, což umožní vozidlu vjet. Když vozidlo poprvé vjede na parkoviště, RFID modul zaznamená informace o vozidle. Při odjezdu vozidla z parkoviště je opět naskenována IC karta pro účtování poplatků. Po zaplacení se závora automaticky otevře, což umožní vozidlu vyjet, čímž se zvyšuje efektivita chytrého parkoviště.

Zobrazení parkovacích míst: Počet vozidel, která vjíždějí, je počítán pomocí fotoelektrických senzorů, zatímco počet vozidel, která vyjíždějí, je zaznamenáván RFID modulem. Poté se na obrazovce zobrazí vypočítaný zbývající počet parkovacích míst spolu s



celkovým počtem parkovacích míst, čímž se předchází problému, kdy majitelé vozidel nemohou najít volná parkovací místa.



Funkce hlasového vysílání: Když vozidlo vjede na parkoviště, přehraje se hlasová zpráva „Vítejte“. Když

vozidlo opustí parkoviště, přehraje se hlasová zpráva „Šťastnou cestu“. Využitím funkce hlasového vysílání na parkovišti mohou řidiči zažít pocit tepla a péče uprostřed rušného městského života.



Laboratoř ITS



Integrovaný chytrý parkovací systém



Popis projektu



Vybudujte komplexní chytrý parkovací systém pro dosažení funkcí, jako je inteligentní přístupová kontrola, zobrazení parkovacích míst a hlasové vysílání.

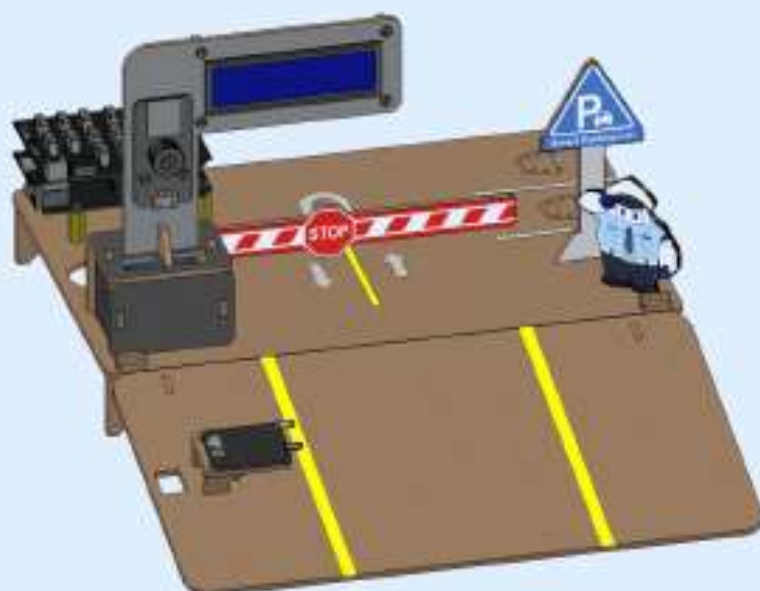


Hardwarové schéma

1. Seznam hardware

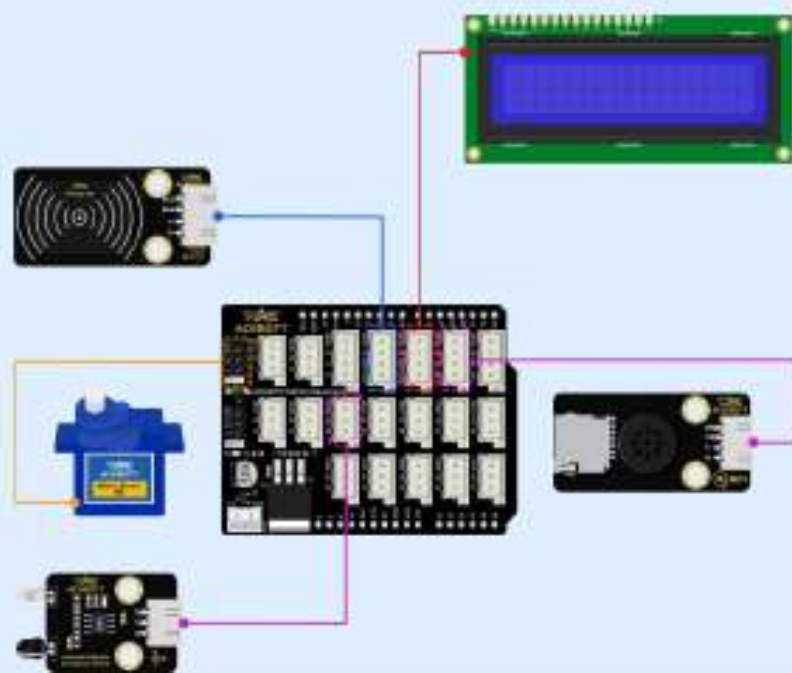
Obrázek	Název	Množství
	ESP32 Board	1
	ESP32 Expansion Board	1
	Servo	1
	Infrared Obstacle Avoidance Sensor	1
	MP3 Module	1
	LCD1602	1
	RFID Module	1
	Type-C Data Cable	1

2. Schéma hardwarové struktury



Podrobné montážní kroky naleznete v oddílu 14 "14 Inteligentní parkovací systém integrován" montážního dokumentu.

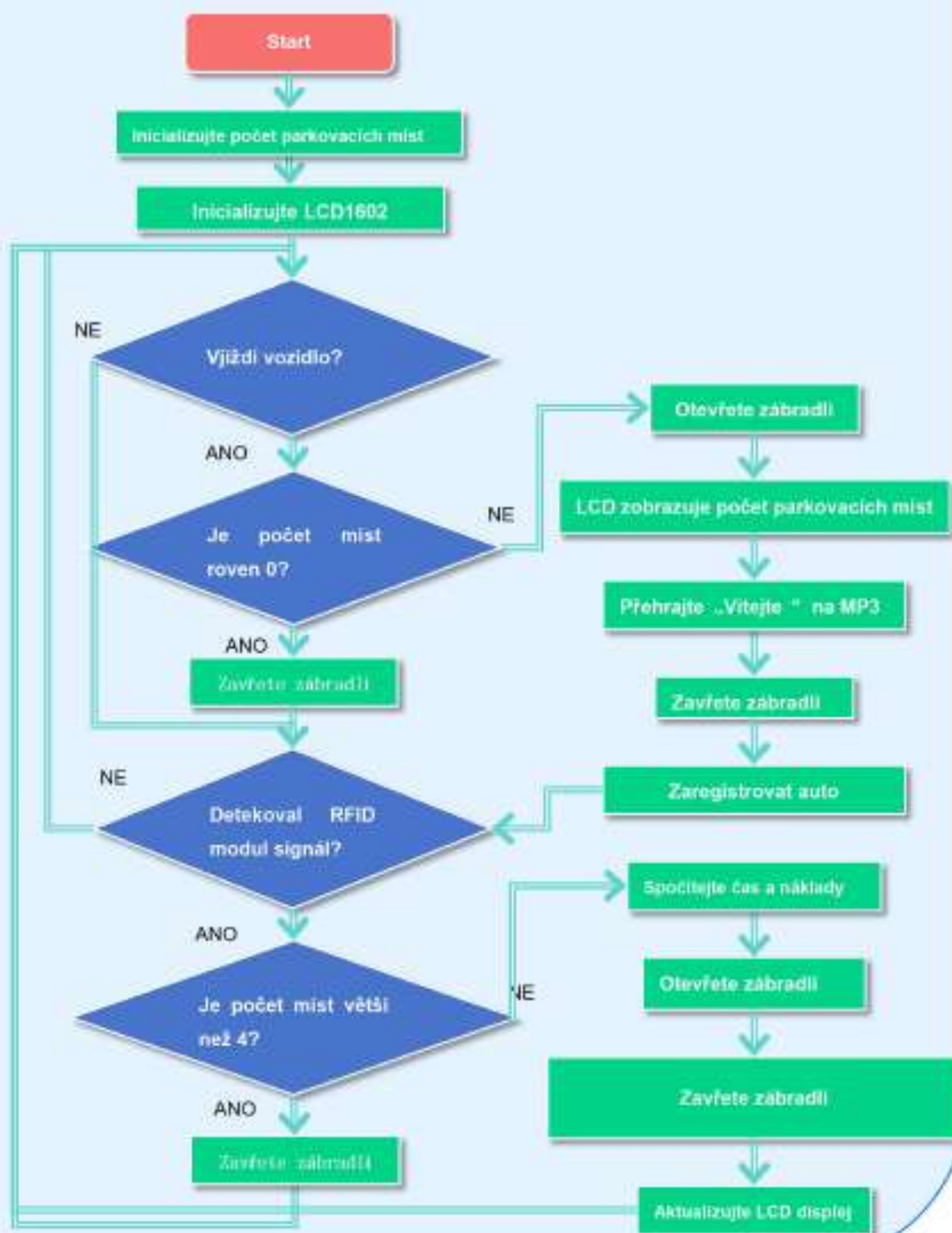
3. Schéma zapojení hardwaru





Programming

1. Programový blokový diagram



2. Kódování

```
#include <ACB_MP3.h> //import MP3 library
#include <Wire.h> //import I2C library
#include <MFRC522_I2C.h> //import RFID library
#include <ESP32_Servo.h> //import servo library
#include <hd44780.h> //import LCD library
#include <hd44780ioClass/hd44780_I2Cexp.h> //import LCD library
#include <List.hpp> //import List library

#define serve 4 //define pin of servo
#define PE 23 //define pin of photoelectric sensor
#define railing_close 90 //set the servo angle
#define railing_open 0 //set the servo angle
#define numRows 2 //declare the rows number of LCD
#define numCols 16 //declare the cols number of LCD
MFRC522_I2C mfrc522(0x28, -1); //creat RFID object
Servo Myservo; //creat servo object
hd44780_I2Cexp lcd(0x27, numRows, numCols); //creat LCD1602 object
ACB_MP3 MyMP3; //creat MP3 object
List<String> card_list; //creat list object for IC card
List<int> time_list; //creat list object for time

int Parking_space = 5; //variable of parking space
int cost = 0; //variable of parking cost
String rfid_str; //variable of RFID

void setup() {
    Wire.begin(); //initialize I2C
    mfrc522.PCD_Init(); //initialize RFID
    Myservo.attach(serve, 500, 2500); //initialize servo
    Myservo.write(railing_close); //close railing
    delay(500);
    lcd.begin(numCols, numRows); //initialize LCD1602
    lcd.backlight(); //turn on backlight of LCD
    lcd.clear(); //clear screen of LCD
    pinMode(PE, INPUT); //set pin of PE sensor as INPUT
    MyMP3.init(16, 17); //initialize MP3
    delay(500);
    Serial.begin(115200);
}
```

```

void loop() {
    spaceDisplay(Parking_space); //display parking space
    if(digitalRead(PE) == 0) { // detect if is there a car coming in
        if(Parking_space == 0) { //detect if is there no space
            Myservo.write(railing_close); //close railing
        } else {
            Myservo.write(railing_open); //open railing
            Serial.println("Please register car information!");
            lcd.clear();
            lcd.setCursor(0, 0); //set cursor of LCD
            lcd.print("Register IC Card"); //display title
            //wait for reading IC card to register car information
            while(!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial());
            MyMP3.playInFolder(0x03); //play prompt tone
            delay(1000);
            rfid_str = "";
            Serial.print(F("Card UID:"));
            for (byte i = 0; i < mfrc522.uid.size; i++) { //read IC card
                rfid_str = rfid_str + String(mfrc522.uid.uidByte[i], HEX);
            }
            //check if the card list already contains the IC card:
            if(card_list.getSize() > 0){
                //check if the card list already contains the IC card:
                while(indexOfList(rfid_str) >= 0){
                    Serial.println("This card already exists. Use new card.");
                    lcd.clear();
                    lcd.setCursor(0, 0); //set cursor
                    lcd.print("Failed!"); //display text
                    lcd.setCursor(0, 1); //set cursor
                    lcd.print("Use New Card!"); //display text
                    //read IC card again:
                    while(!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial());
                    MyMP3.playInFolder(0x03); //paly prompt tone
                    delay(1000);
                    rfid_str = "";
                    Serial.print(F("Card UID:"));
                    for (byte i = 0; i < mfrc522.uid.size; i++) { //read IC card
                        rfid_str = rfid_str + String(mfrc522.uid.uidByte[i], HEX);
                    }
                }
            }
        }
    }
}

```

```

    }
  }
}
card_list.add(rfid_str);
MyMP3.playInFolder(0x01); //play "welcom"
delay(2000); //delay 2s
Parking_space -= 1; //decrease parking space by 1
MyServo.write(railing_close); //close railing
time_list.add(millis()); //store time of register
lcd.clear();
Serial.println("Successful! Start billing.");
lcd.setCursor(0, 0); //set cursor
lcd.print("Successful!"); //display text
lcd.setCursor(0, 1); //set cursor
lcd.print("Start Billing!"); //display text
delay(2000);
lcd.clear(); //clear screen
}
}
//Check if a vehicle is exiting.
if(!mfrc522.PICC_IsNewCardPresent()
|| !mfrc522.PICC_ReadCardSerial()) {
  delay(50);
  return;
}
rfid_str = "";
Serial.print(F("Card UID:"));
for (byte i = 0; i < mfrc522.uid.size; i++) { //read IC card ID
  rfid_str = rfid_str + String(mfrc522.uid.uidByte[i], HEX);
}
Serial.println("Billing...");
lcd.clear();
lcd.setCursor(0, 0); //set cursor
lcd.print("Billing..."); //display text
delay(1000);
//timing and billing, 2$ per second
cost = ((millis() - time_list.indexOfList(rfid_str)) / 1000) * 2;
time_list.remove(indexOfList(rfid_str)); //remove the ID in list
card_list.remove(indexOfList(rfid_str)); //remove the time in list
Serial.println(cost);

```

```

lcd.setCursor(0, 1); //set Cursor
lcd.print("Cost:"); //display test
lcd.setCursor(5, 1); //set Cursor
lcd.print(cost); //display cost
delay(1000);
MyServo.write(railing_open); //open railing
MyMP3.playInFolder(0x02); //play "save journey"
delay(2000);
MyServo.write(railing_close); //close railing
Parking_space += 1; //increase parking space by 1
lcd.clear();//clear screen
}

int indexOfList(String s){//get position of element in list
  for(int i = 0;i < card_list.getSize();i++){
    if(rfid_str == card_list[i]){
      Serial.println(i);
      return i;
    }
  }
  return -1;
}

void spaceDisplay(int Parking_space){//display parking space
  lcd.setCursor(0, 0); //set cursor
  lcd.print("Parking Space"); //display text
  lcd.setCursor(0, 1); //set cursor
  lcd.print("Total:"); ////display "Total"
  lcd.setCursor(6, 1); //display text
  lcd.print(5); //display total space
  lcd.setCursor(8, 1); //set cursor
  lcd.print("Rest:"); //display "Rest"
  lcd.setCursor(13, 1); //set cursor
  lcd.print(Parking_space); //display rest space
  delay(200);
}

```

3. Vysvětlení instrukcí

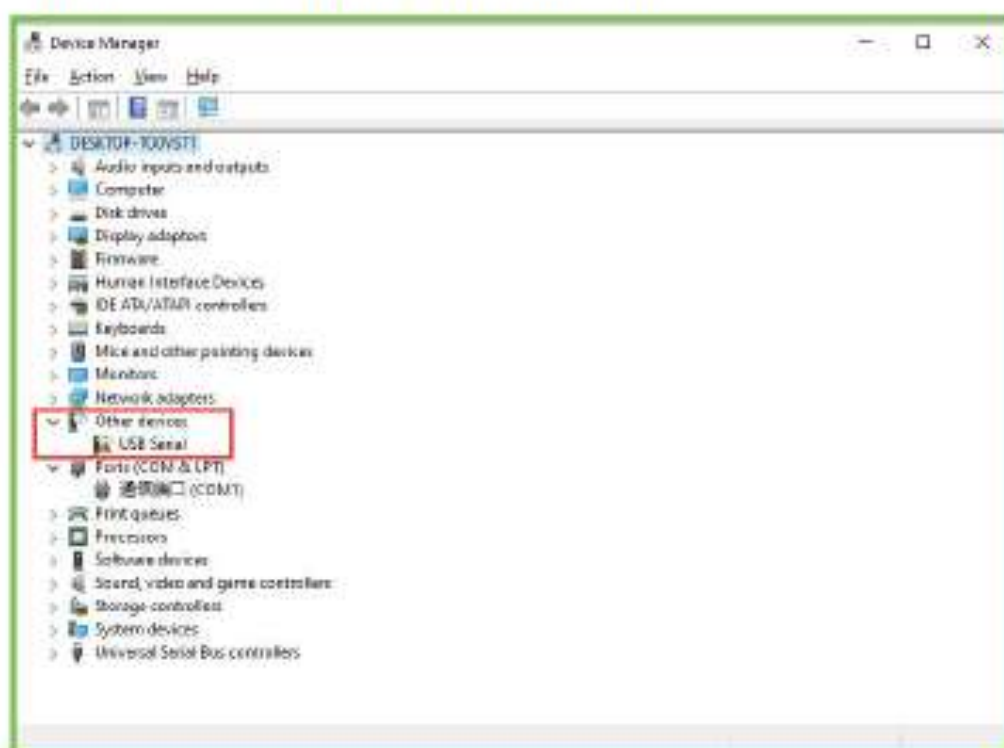
<code>List.hpp</code>	Seznam knihoven v Arduino.
<code>List<String> list</code>	Vytvořte objekt seznamu s parametrem uvnitř ostrých závorek, který určuje datový typ prvků seznamu.
<code>list.add(e)</code>	Přidejte specifikovaný prvek na konec seznamu.
<code>list.getSize()</code>	Získejte počet prvků v seznamu.
<code>list.remove(p)</code>	Odstraňte prvek na specifikované pozici v seznamu.

obrázku



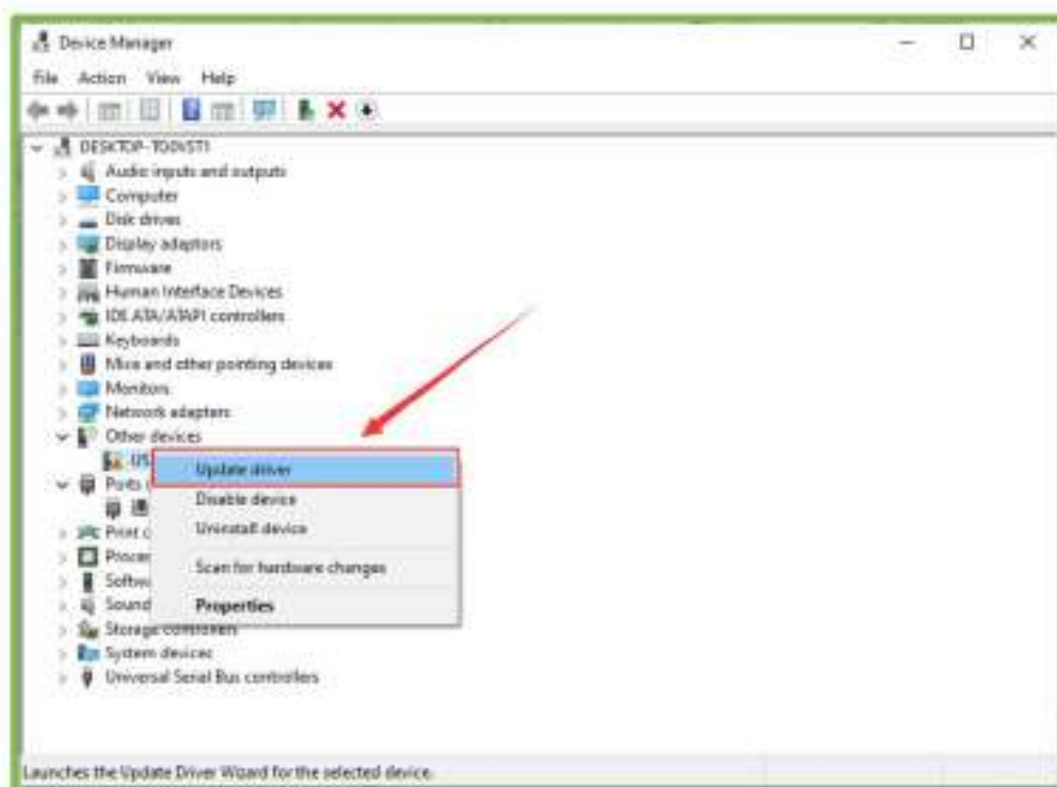
Řešení problému s instalací ovladače

Pokud instalace ovladače během instalace Arduino IDE selže, po prvním připojení desky ESP32 k počítači klikněte pravým tlačítkem na 'My Computer' -> klikněte na 'Properties' -> klikněte na 'Device Manager'. V sekci 'Other devices' uvidíte 'USB-Serial' nebo 'Unknown Device'.

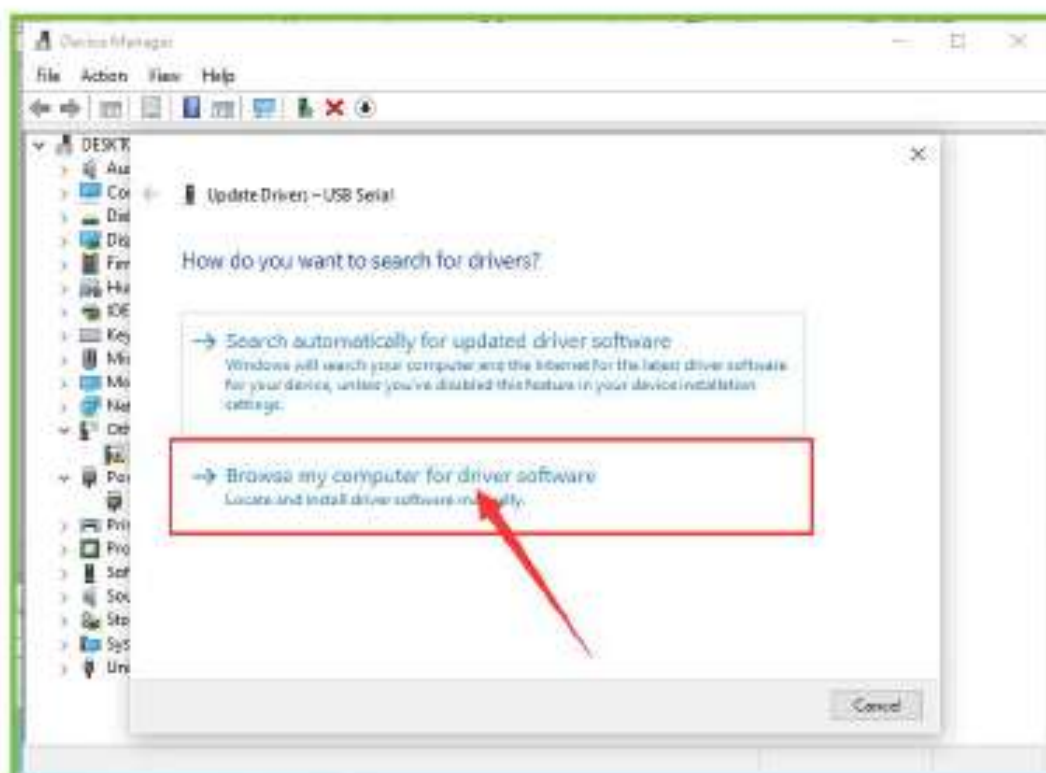


V tuto chvíli budete potřebovat místní soubor ovladače a manuálně nainstalovat ovladač. Konkrétní kroky jsou následující:

Krok 1: Klikněte pravým tlačítkem na zařízení a vyberte možnost v horním menu (Aktualizovat software ovladače), jak je znázorněno na následujícím



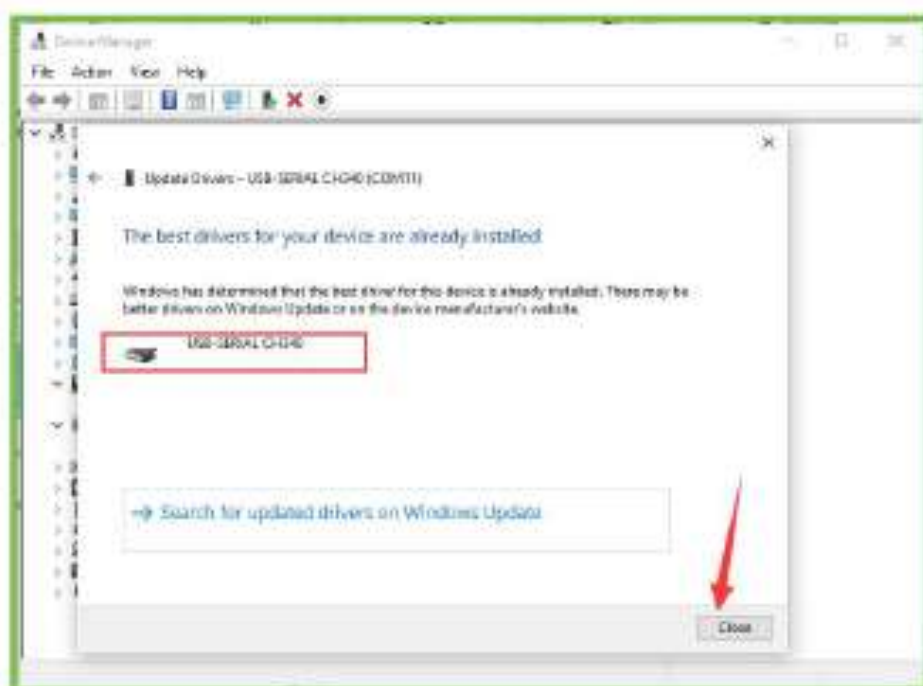
Krok 2: Poté se vám zobrazí možnosti 'Search automatically for updated driver software' nebo 'Browse my computer for driver software', jak je znázorněno na následujícím obrázku. Na této stránce vyberte možnost 'Browse my computer for driver software'.



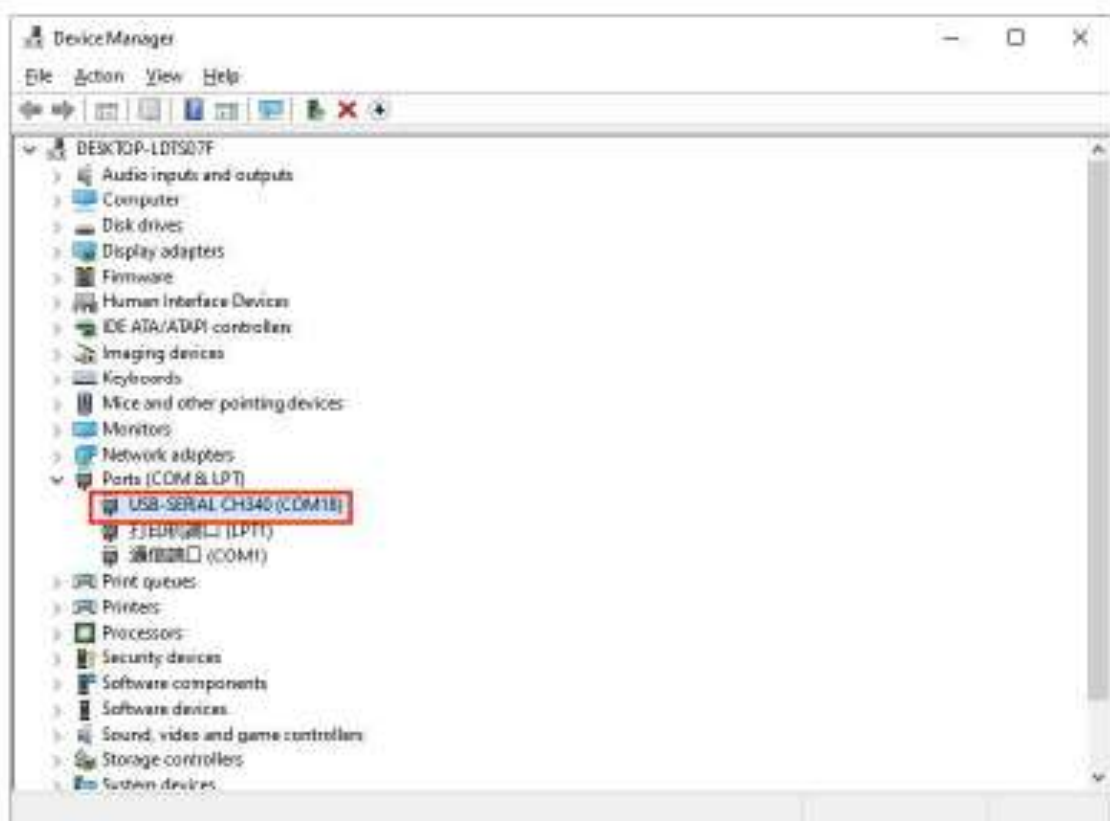
Krok 3: Klikněte na tlačítko 'Browse', přejděte do složek a najděte umístění souboru ovladače v počítači, klikněte na 'OK' a nakonec klikněte na 'Next' pro instalaci ovladače.



Krok 4: Po dokončení instalace ovladače obdržíte potvrzovací zprávu. Jakmile je instalace dokončena, klikněte na 'Close'.



Krok 5: Potvrďte, že ovladač CH340 byl úspěšně nainstalován. Vložte jeden konec USB kabelu do desky ESP32 a druhý konec do USB portu počítače. Klikněte pravým tlačítkem na 'My Computer' -> klikněte na 'Properties' -> klikněte na 'Device Manager'. Po připojení řídicí desky by se mělo objevit následující zobrazení, které potvrzuje, že instalace byla úspěšná.



Sleduj nás

Naskenujte QR kódy a sledujte nás pro odstraňování problémů a nejnovější zprávy.

Máme velmi rozsáhlou komunitu, která je velmi nápomocná při odstraňování problémů, a máme také tým podpory připravený zodpovědět jakékoli dotazy.



QR kód ACEBOTT FB QR kód YouTube



ACEBOTT STEM Education Tech Co.,Ltd

0755-23956813

www.acebott.com

Room 306, Building 212, Tairan Science Park, Terra 4th Rd, Futian District, ShenZhen, China, 518042