

Education  
Solution Series

# Smart Home Education Kit Level 1





# Obsah

|  |     |
|--|-----|
| Před použitím je nutné si přečíst..... | 3   |
| Způsob instalace ovladače .....        | 5   |
| Stavební kroky.....                    | 15  |
| 01 Rozsvícení LED diody .....          | 16  |
| 02 Pulzující světla.....               | 21  |
| 03 Tlačítko Light .....                | 22  |
| 04 Dotykové světlo .....               | 25  |
| 05 Inteligentní světlo.....            | 28  |
| 06 RGB světlo.....                     | 32  |
| 07 Elektronické kostky .....           | 35  |
| 08 Elektronický bzučák .....           | 40  |
| 09 Inteligentní přehrávač hudby.....   | 44  |
| 10 Barevná světla .....                | 50  |
| 11 Hudební světlo.....                 | 54  |
| 12 Inteligentní detekce T&H.....       | 63  |
| 13 Inteligentní ventilátor.....        | 67  |
| 14 Inteligentní řídicí systém T&H..... | 70  |
| Kapitola 1 Budoucí domov.....          | 75  |
| Oddíl 1 Chytrá domácnost .....         | 76  |
| Oddíl 2 Mozek chytré domácnosti.....   | 85  |
| Kapitola 2 Rozsviňte svůj domov.....   | 108 |
| Oddíl 1 Rozsvícení světla .....        | 109 |
| Oddíl 2 Pulzující světla .....         | 119 |
| Oddíl 3 Řízení světla (I) .....        | 131 |
| Oddíl 4 Řízení světla (II).....        | 141 |





|   |            |
|---|------------|
| Oddíl 5 Chytré světlo .....                   | 147        |
| <b>Kapitola 3 Šťastný domov.....</b>          | <b>156</b> |
| Oddíl 1 Duhový dům .....                      | 157        |
| Oddíl 2 Elektronické kostky.....              | 165        |
| Oddíl 3 Elektronický bzučák.....              | 179        |
| Oddíl 4 Chytrý hudební přehrávač.....         | 189        |
| Oddíl 5 Hudební světlo (I).....               | 200        |
| Oddíl 6 Hudební světlo (II).....              | 207        |
| <b>Kapitola 4 Teplý a pohodlný domov.....</b> | <b>216</b> |
| Oddíl 1 Detekce teploty a vlhkosti.....       | 217        |
| Oddíl 2 Chytrý ventilátor .....               | 224        |
| Oddíl 3 Regulace teploty a vlhkosti.....      | 232        |
| <b>Často kladené otázky .....</b>             | <b>242</b> |
| <b>Sledujte nás.....</b>                      | <b>246</b> |





# Před použitím je nutné si přečíst

## 1. Upravená baterie

Z důvodu omezení přepravy nemůžeme s produktem dodat požadované baterie. Omlouváme se za případné nepříjemnosti. Abychom vám pomohli rychle najít vhodné baterie, sestavili jsme pro vás několik doporučených odkazů na Amazonu, včetně baterií 18650 (pro napájení chytré domácnosti), nabíječek (pro nabíjení baterií 18650) pro vaši orientaci.

- (1) Baterie 18650 Sharp Odkaz : <https://www.amazon.com/Rechargeable-Originalní18650-Svítilny-Svítilny na dveře-Svítilny na hlavu/dp/B0CWX67HTD>
- (2) Odkaz na nabíječku : <https://shop.cebott.com/products/cebott-us-plug-eu-plug-%20uk-plug-dual-charger-for-18650-14500-16430-rechargeable-li-ion-battery?vari%20ant=4299>

**Poznámka: Tento výrobek musí být vybaven baterií 18650 Sharp-Ended \*2; nabíječkou \*1.**

## 2. Poprodejní podpora

Pokud se setkáte s jakýmkoli problémem, kontaktujte prosím náš tým podpory prostřednictvím e-mailu na adrese [support@cebott.com](mailto:support@cebott.com), na který vám odpovíme do 24 hodin. Můžete také naskenovat níže uvedený QR kód a sledovat nás, abychom vám poskytli návody na řešení problémů a nejnovější aktualizace.



FB skupina ACEBOTT QR kód



QR kód YouTube





### 3. Video s instalací softwaru

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

Instalace ovladače sériového portu CH340 : <https://youtu.be/B0afbif4jXA>

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

Instalace knihoven : <https://youtu.be/BIKFr2oVEVQ>





# Způsob instalace ovladače

Sériový port je také druh počítačového komunikačního rozhraní, obvykle se používá pro počítač a další zařízení (např. modem, senzor, tiskárna, mikro kontrolér atd.) pro přenos dat, sériový port USB je naše nejčastěji používané komunikační rozhraní.

Obecně platí, že po počáteční instalaci softwaru Arduino IDE je třeba nainstalovat jednotku sériového portu pro přenos upraveného programu na řídicí desku.

Připojte řídicí desku k počítači pomocí kabelu USB a ovladače se automaticky nainstalují do systémů MacOS a Windows. Pokud se instalace ovladače nezdaří, je třeba ovladač nainstalovat ručně.

Tento návod obsahuje dva způsoby instalace ovladačů v systémech Windows a MacOS.

## I. Nainstalujte ovladač sériového portu CH340 do systému Windows

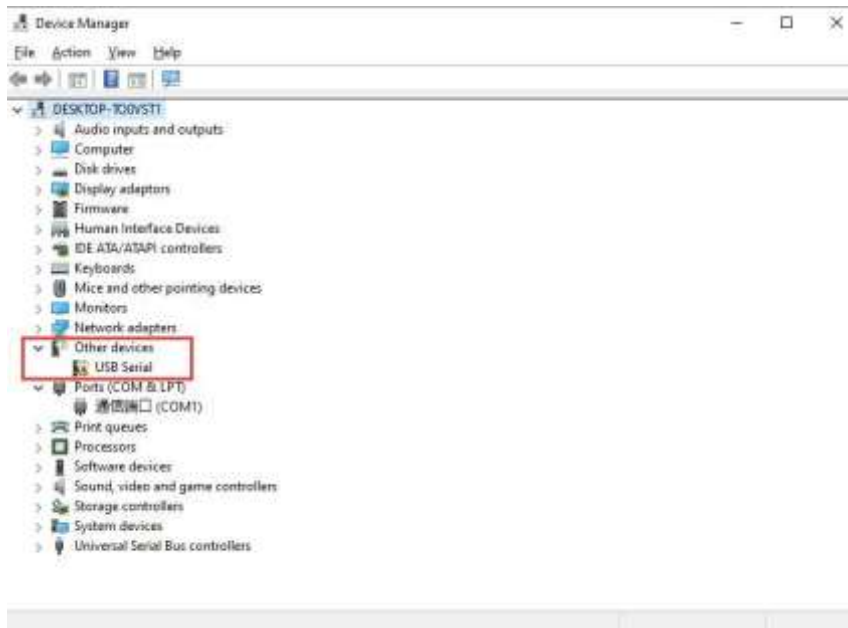
Instalaci ovladače lze stáhnout online nebo můžete najít složku s názvem "USB\_Drive\_CH341\_3\_1" v námi poskytnutém balíčku zdrojů, vybrat odpovídající instalační balíček podle typu počítačového systému a poté jej nainstalovat podle následujících kroků.

### 1. Zkontrolujte, zda je nainstalována jednotka sériového portu CH340 (pokud je nainstalována, přeskočte).

① Vložte jeden konec kabelu USB do řídicí desky ESP32 a druhý konec do rozhraní USB v počítači.

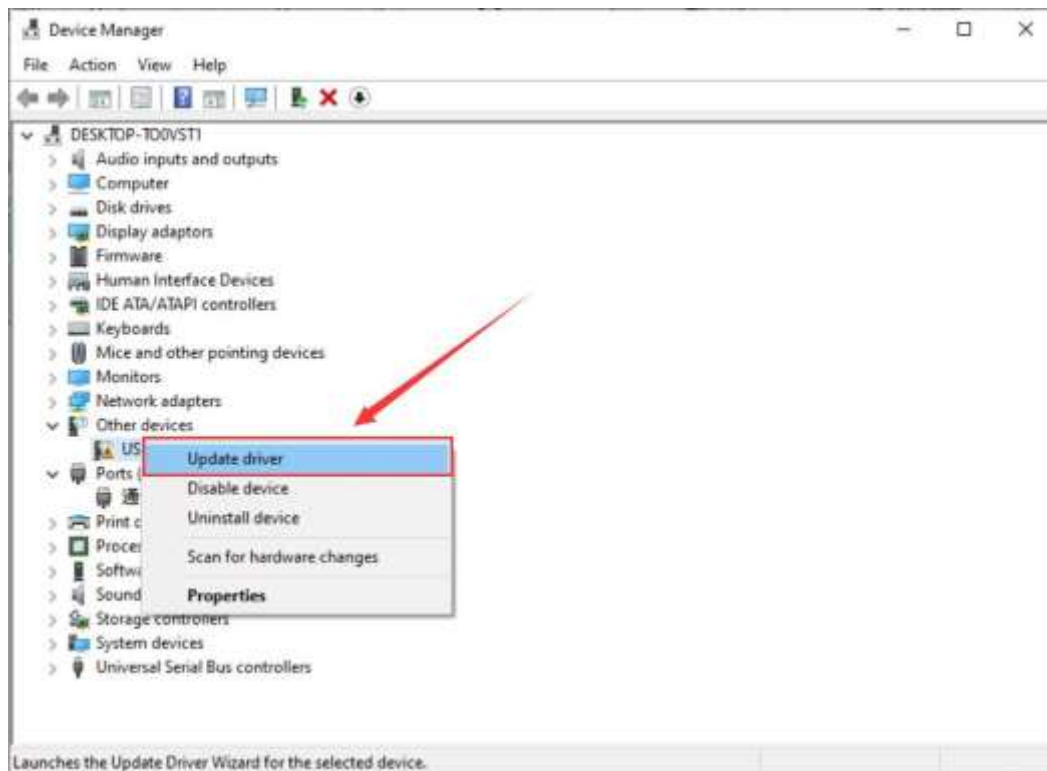
② Při prvním připojení řídicí desky ESP32 k počítači klikněte pravým tlačítkem myši na "Můj počítač" -> "Vlastnosti" -> klikněte na "Správce zařízení" a v části "Další zařízení" uvidíte "USB-Serial" nebo "Neznámé zařízení".





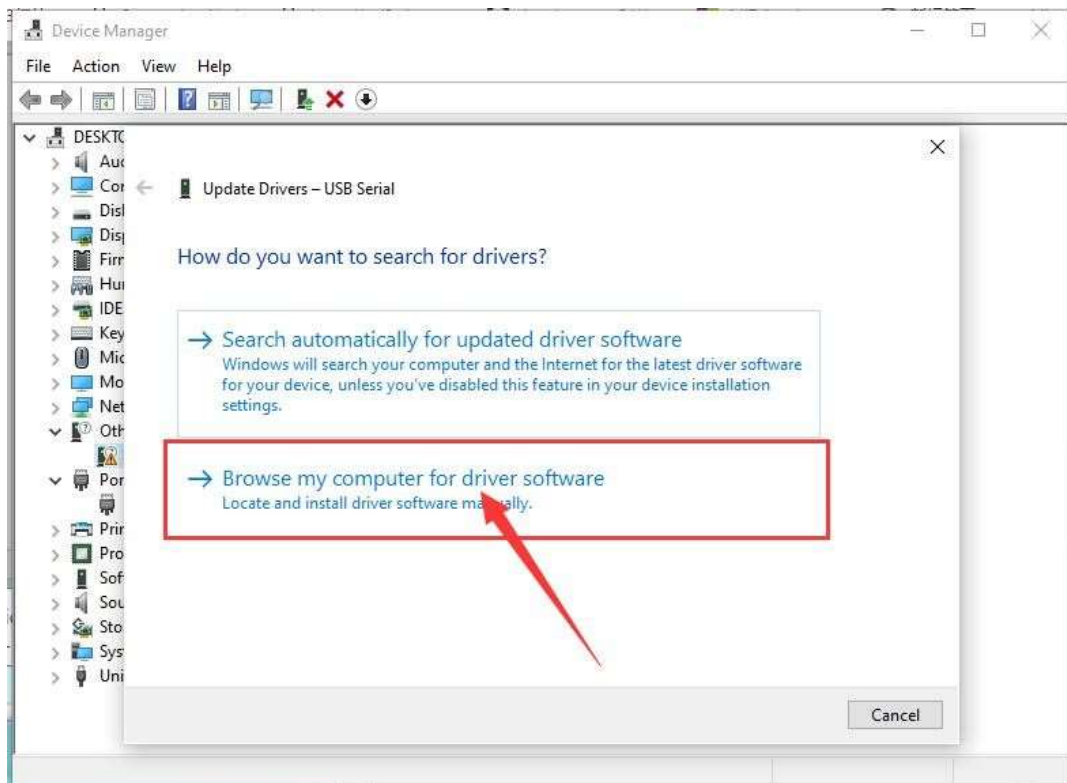
## 2. Instalace jednotky sériového portu CH340

① Klikněte pravým tlačítkem myši na zařízení a vyberte možnost v horní nabídce (Aktualizace softwaru ovladače), jak je znázorněno na obrázku níže.

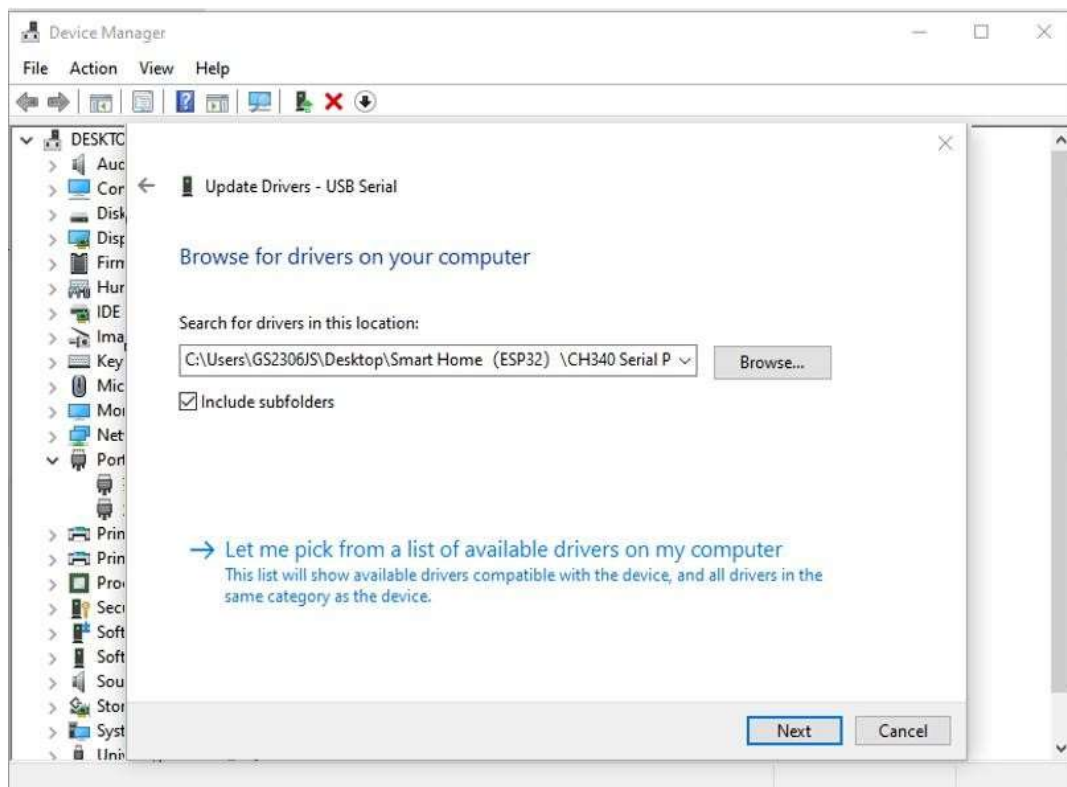


② Poté se zobrazí výzva k "automatickému vyhledání aktualizovaného softwaru ovladače" nebo "procházení mého počítače pro nalezení softwaru ovladače", jak je znázorněno níže, na této stránce vyberte "Procházet můj počítač pro nalezení softwaru ovladače".

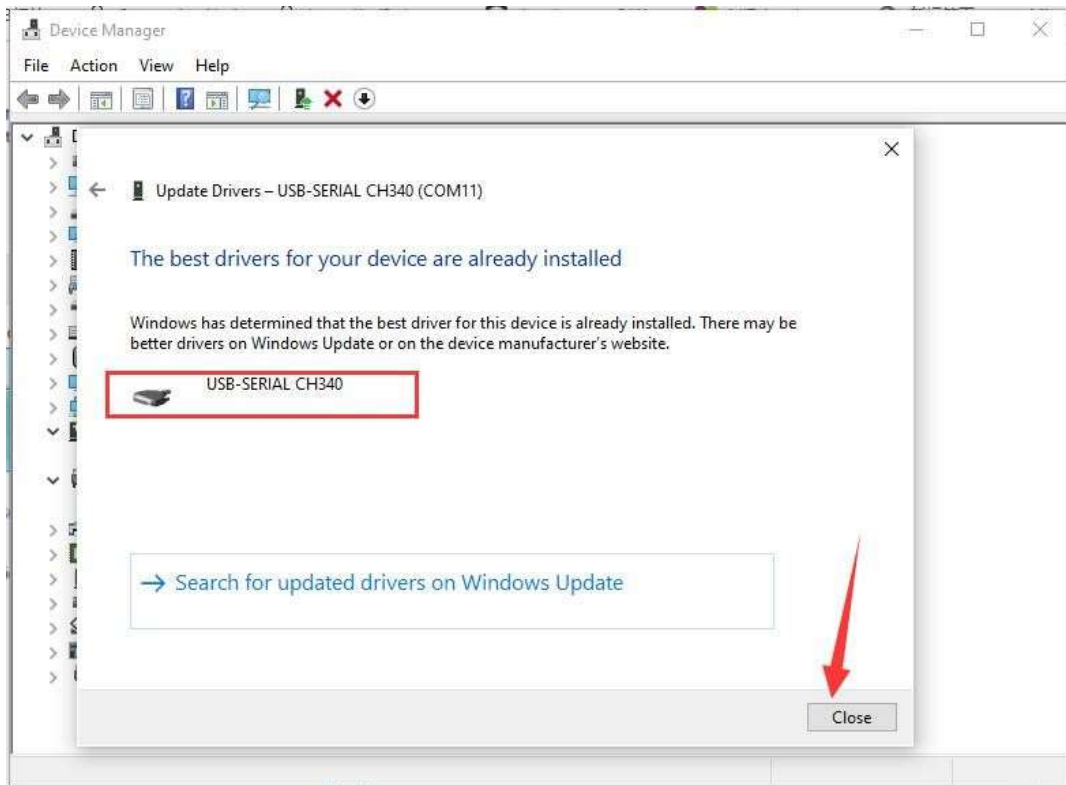




③ Poté přidejte cestu k souboru ovladače.

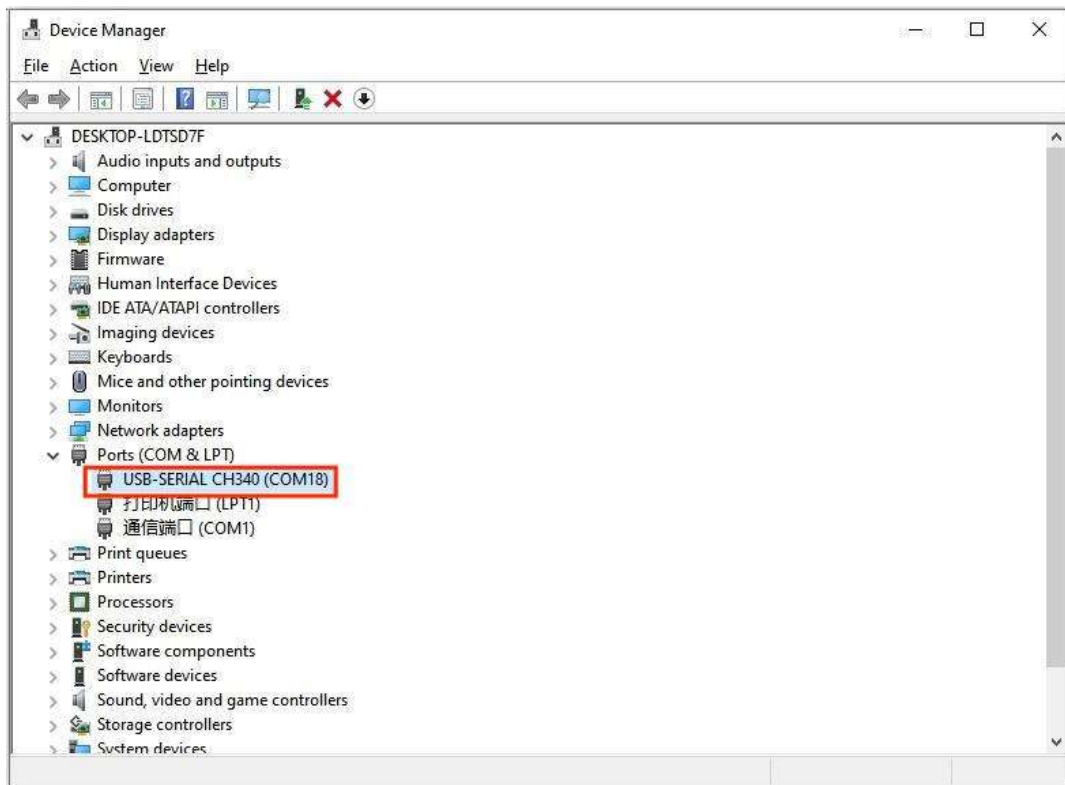


④ Po dokončení instalace softwaru se zobrazí potvrzovací zpráva a poté klikněte na tlačítko "Zavřít".



### 3. Ověřte, zda byla jednotka CH340 se sériovým portem úspěšně nainstalována.

Vložte jeden konec kabelu USB do řídicí desky ESP32 a druhý konec do rozhraní USB v počítači. Klikněte pravým tlačítkem myši na "Můj počítač" -> "Vlastnosti" -> klikněte na Správce zařízení a po připojení k horní desce řadiče je instalace úspěšná.



## II. Instalace ovladače sériového portu CH340 v systému MacOS

Otevřete složku "CH340 Driver File-MAC" a zobrazte instalační soubor ovladače, který poskytujeme. Obecně se ve výchozím nastavení instalují ovladače ve formátu pkg, a pokud systém OS X 11.0 nebo novější nepodporuje Rosettu, nainstalujte ovladače ve formátu dmg.



CH34xVCPDriver.dmg



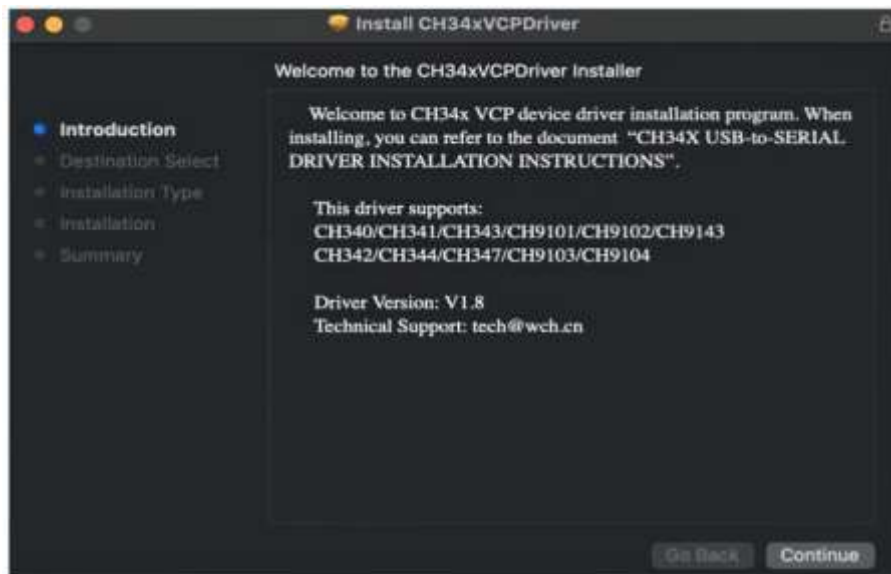
CH34xVCPDriver.pkg

Před instalací přejděte do "Předvolby systému" -> "Zabezpečení a soukromí" -> "Obecná stránka pod názvem " vyberte-> "Mac App Store a identifikovaný vývojář", aby ovladač fungoval.



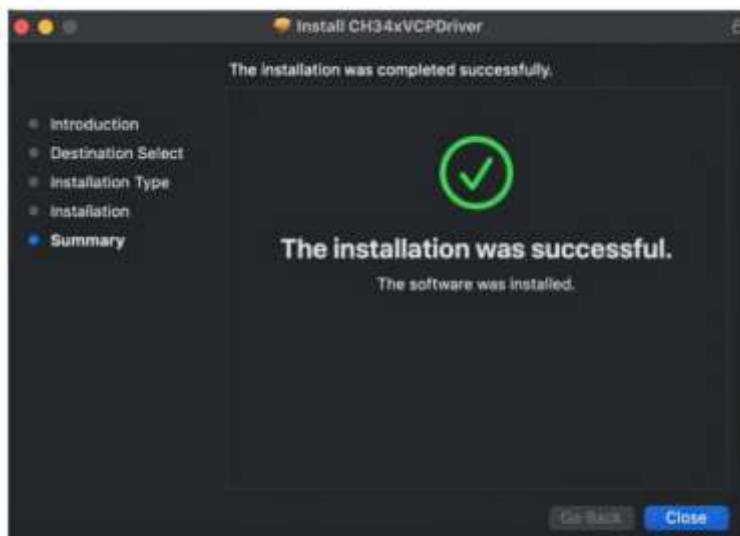
## 1. Instalace ovladačů ve formátu pkg

① Klikněte na soubor ovladače-> Pokračovat-> Nainstalovat.





② úspěšně nainstalován.





③ Instalace ovladače ve formátu pkg v OS X 11.0 a novějším: otevřete "LaunchPad" -> "CH34xVCPDriver" -> Instalovat.



④ Při používání OS X 10.9 až OS X 10.15 restartujte počítač kliknutím na tlačítko "Restartovat" a po restartu proveďte následující kroky.



**2.Nainstalujte ovladač formátu dmg (Pozor: Pokud byl krok 1 úspěšně nainstalován, tento krok přeskočte.)**

① Instalace dmg ovladače, klikněte na dmg soubor a přetáhněte "CH34xVCPDriver" složky aplikací operačního systému.





②Poté otevřete "LaunchPad" -> "CH34xVCPDriver" -> instalace.



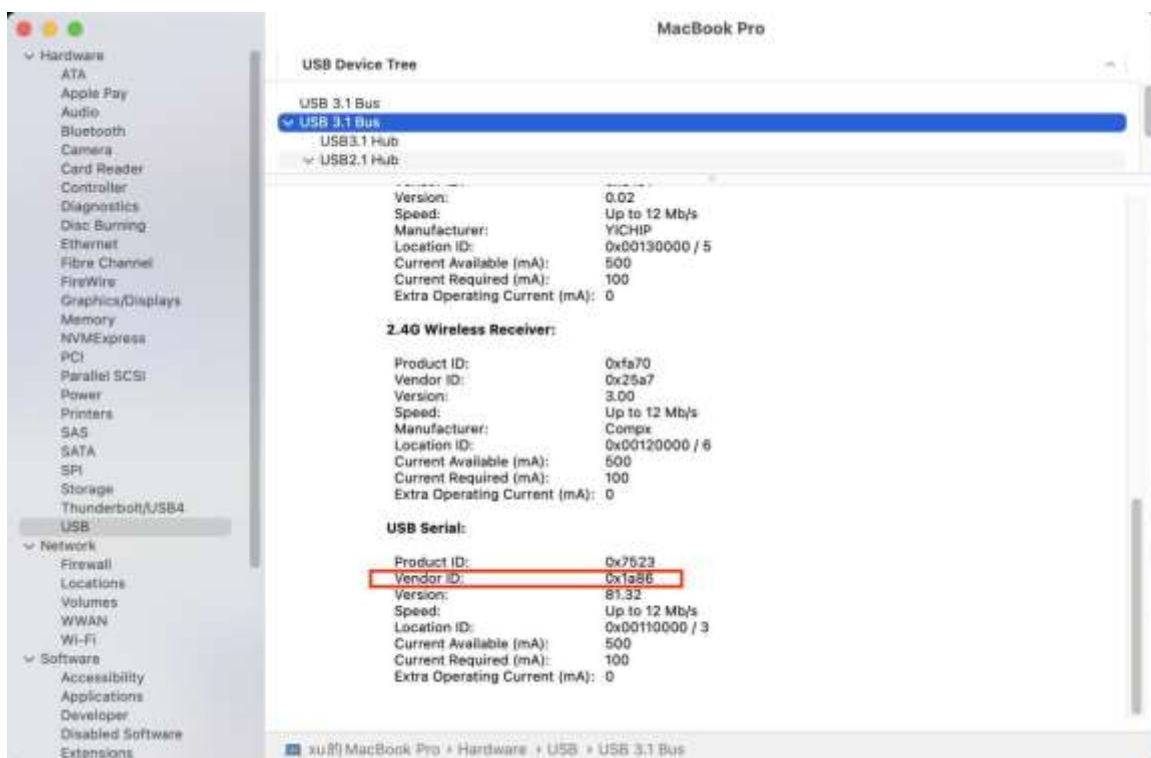


③ úspěšně nainstalován.



### 3. Zkontrolujte, zda je jednotka sériového portu CH340 plně nainstalována.

Po vložení palubní desky do portu USB otevřete Systémová zpráva-> Hardware-> USB. Zařízení USB. Pokud zařízení USB funguje správně, lze nalézt zařízení s "Vendor ID" [0x1a86].





# ACEBOTT

Sada EDU Smart Home Level 1  
**Stavební kroky**





Poznámka:

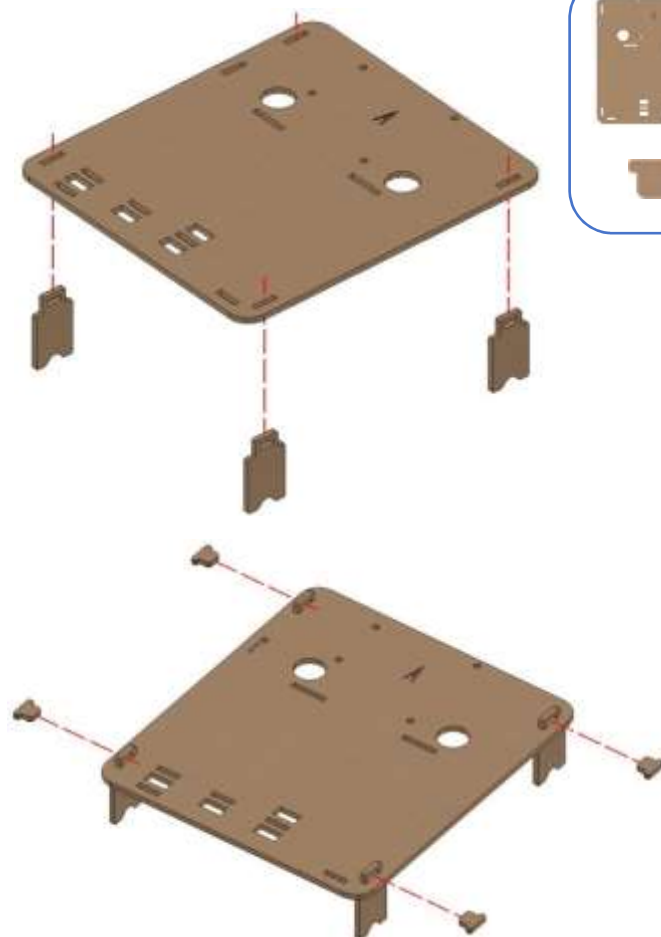
1. Pokud nelze otevřít hypertextový odkaz na program v dokumentu, musíte najít cestu "English\Arduino\3.Program" ve stažených výukových zdrojích a otevřít příslušný program.





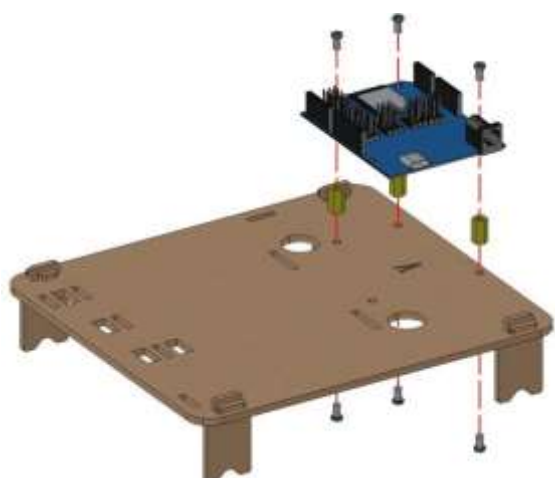
## 01 Rozsvícení LED diody

Krok 1: Instalace základny.



Poznámka: Ujistěte se, že strana dřevěné desky s písmenem "A" směřuje nahoru.

Krok 2: Instalace řídicí desky ESP 32.

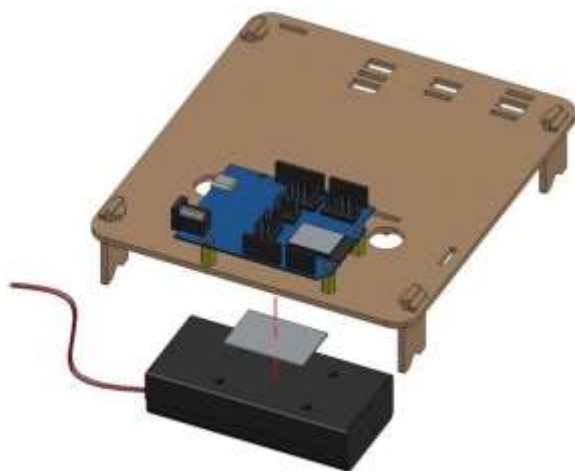


Poznámka: Nejprve připevněte měděný sloupek na základnu a poté nainstalujte ESP.32 Řídicí deska na měděném sloupku.



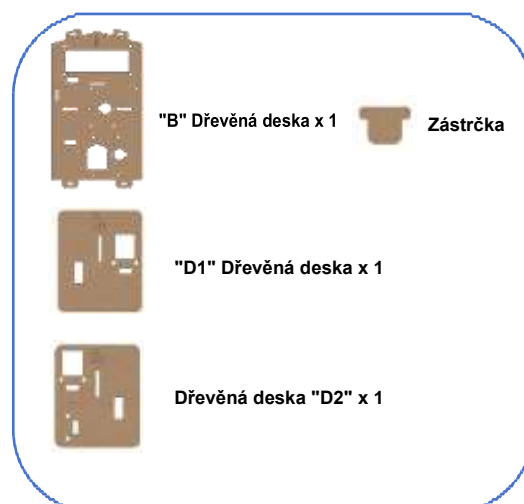
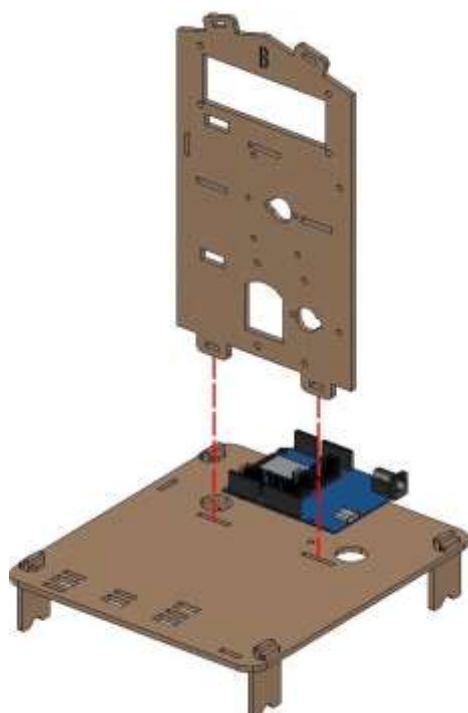


### Krok 3: Instalace držáku baterie.



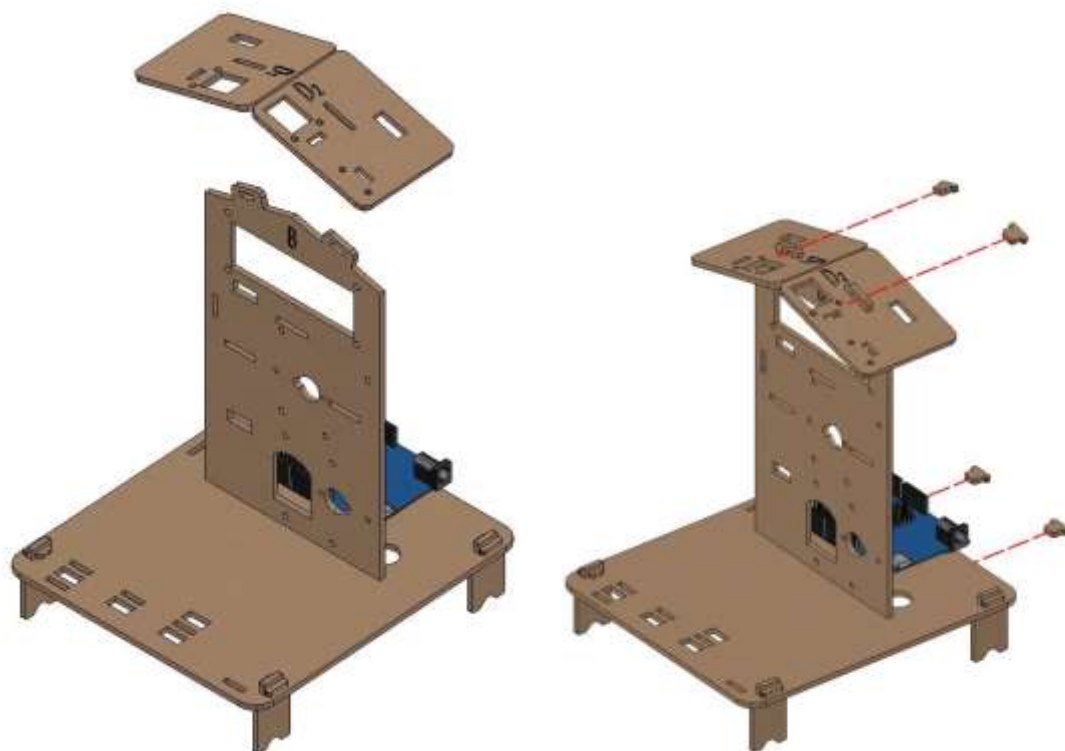
Poznámka: K připevnění držáku baterie na spodní stranu základny použijte oboustrannou pásku a nezakrývejte elektrický vodič a otvory pro kolíky základny.

### Krok 4: Instalace dřevěné desky.



Poznámka: Ujistěte se, že strana dřevěné desky s písmenem "B" směřuje dopředu.





Poznámka: Ujistěte se, že strana dřevěné desky s písmenem "D1" nebo "D2" směřuje nahoru.

Krok 5: Instalace modulu LED.



Bílý LED modul x 1



Šrouby s kulatou hlavou M3\*8MM x



Niklované matice M3 x 2

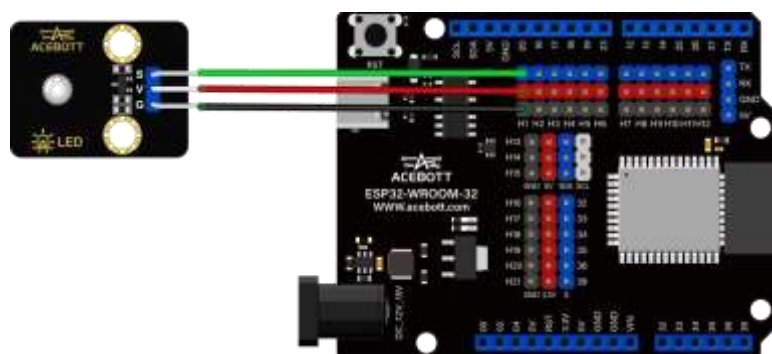




Krok 6: Upevnění stojící karty na základnu.



Krok 7: Zapojení.





## Pokyny pro zapojení:

| Název modulu   | Vývod modulu | Vývod ESP32 |
|----------------|--------------|-------------|
| Bílý LED modul | S            | pin5        |
|                | V            | 5V          |
|                | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)





## 02 Pulzující světla

**Žhavá výzva: Struktura projektu 2 je stejná jako u projektu 1 (zobrazeno níže). Viz stavební kroky projektu 1.**



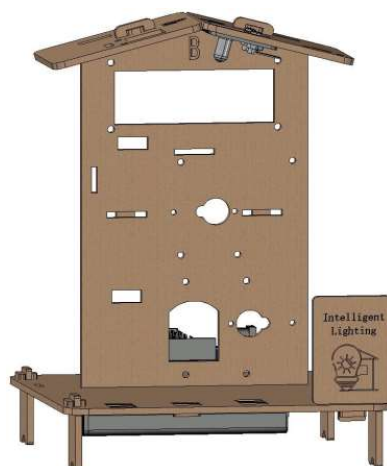
[Kliknutím sem otevřete program pro tento projekt.](#)



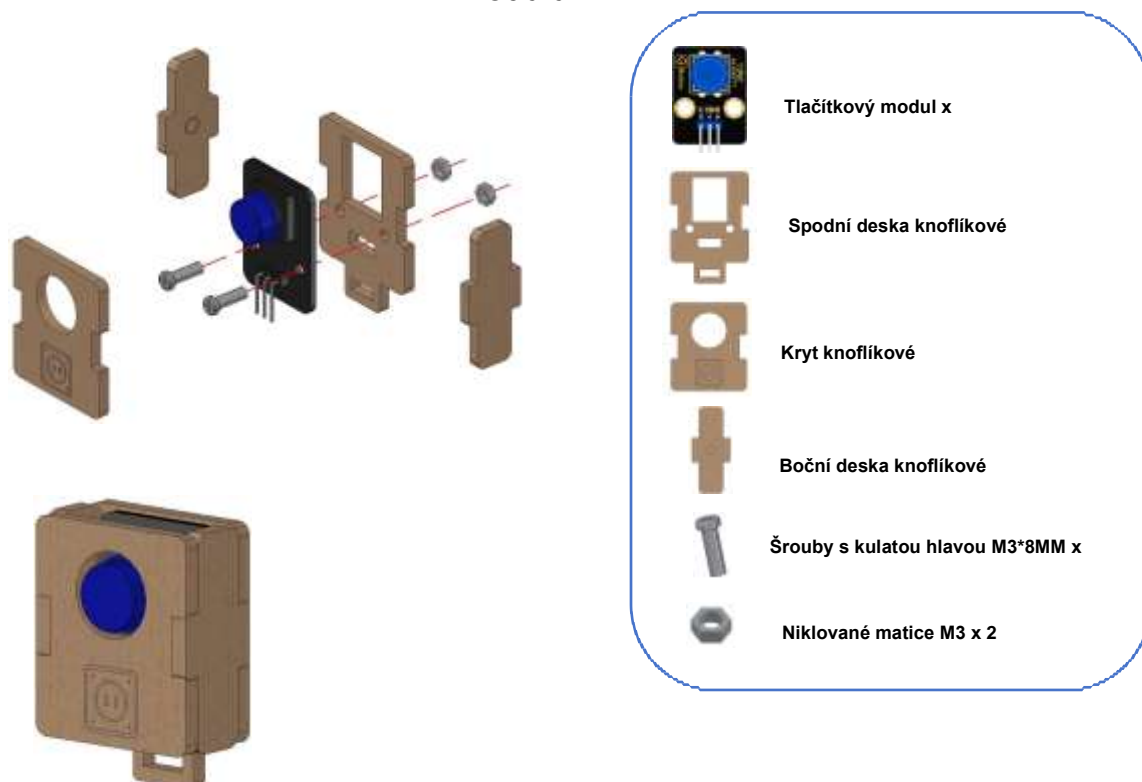


### 03 Tlačítko Light

**Žhavá výzva: Struktura instalace LED v tomto projektu se bude řídit strukturou projektu 1 (zobrazena níže). V této části se můžete odkázat na Projekt 1. Začneme instalací tlačítkového modulu.**



Krok 1: Instalace tlačítkového modulu.

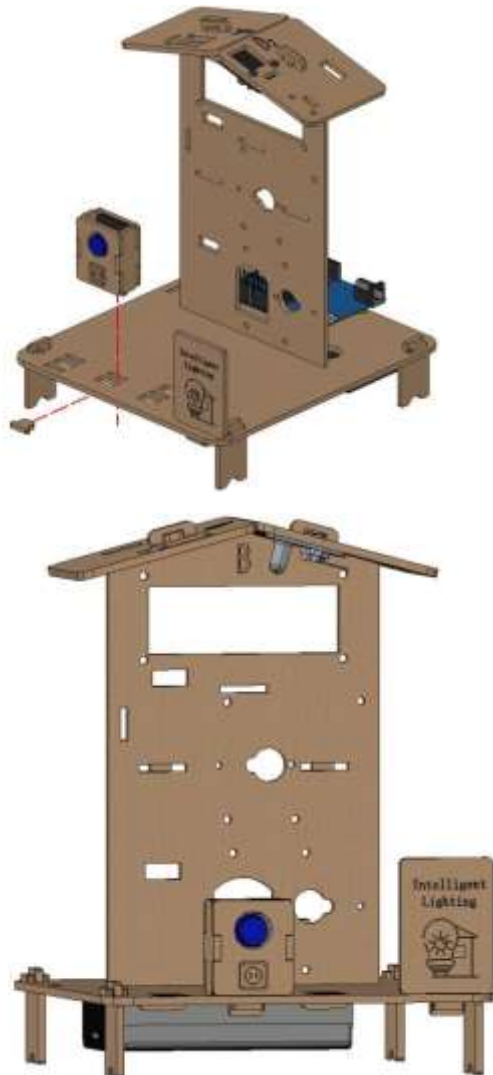


**Poznámka: Nejprve nainstalujte tlačítkový modul na spodní desku pomocí šrouby, poté nainstalujte obě boční desky a nakonec nainstalujte kryt.**



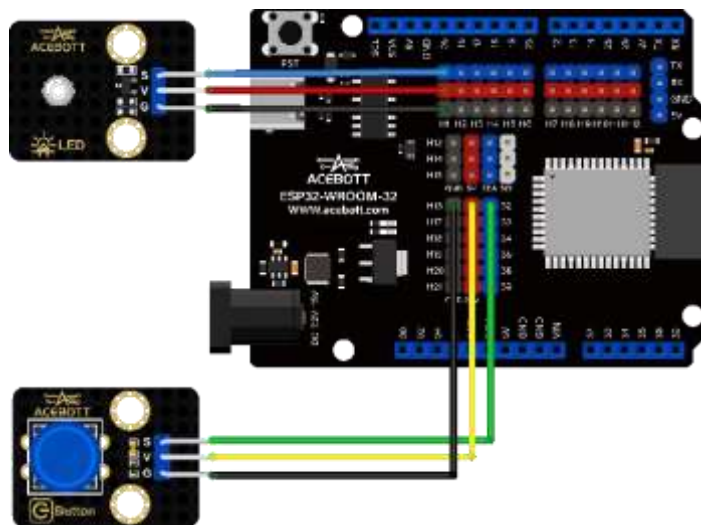


Krok 2: Upevnění tlačítkového modulu na základnu.



Zástrčka x 1

Krok 3: Zapojení.





## Pokyny pro zapojení:

| Název modulu     | Vývod modulu | Vývod ESP32 |
|------------------|--------------|-------------|
| Bílý LED modul   | S            | pin5        |
|                  | V            | 5V          |
|                  | G            | GND         |
| Tlačítkový modul | S            | pin32       |
|                  | V            | 5V          |
|                  | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)



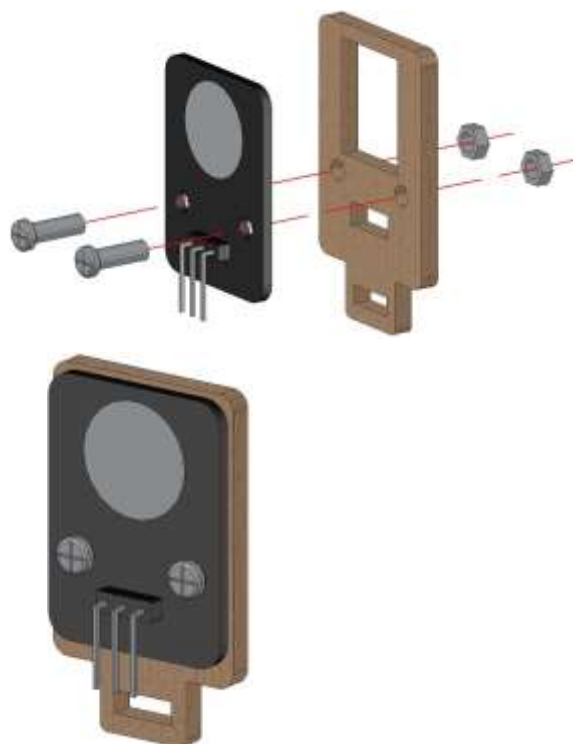


## 04 Dotykové světlo

**Žhavá výzva: Struktura instalace LED v tomto projektu se bude řídit strukturou projektu 1 (zobrazena níže). Konkrétní kroky konstrukce zde nebudou uvedeny. Začneme instalací dotykového senzoru.**



**Krok 1:** Instalace dotykového senzoru.



Dotykový senzor x



Spodní deska senzoru x1



Šrouby s kulatou hlavou M3\*8MM x



Niklované matice M3 x 2

**Krok 2:** Upevnění dotykového snímače na základnu.

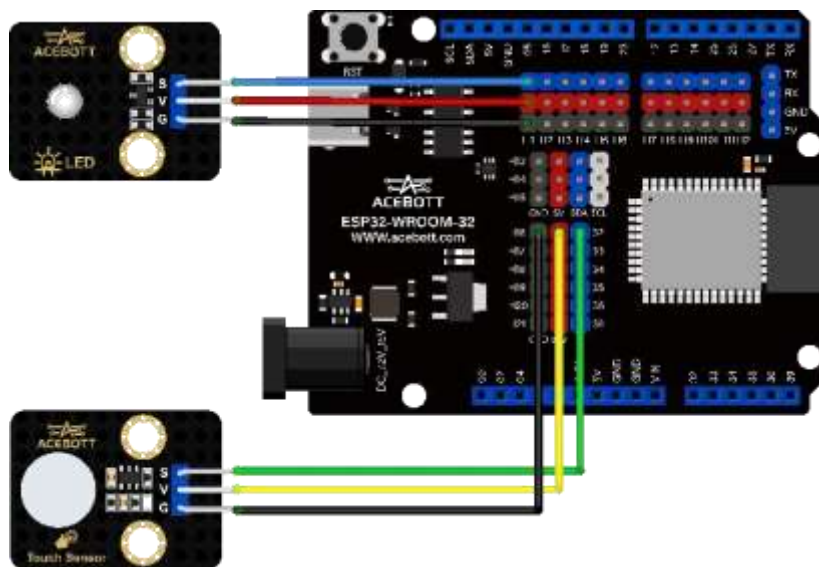




Zástrčka x 1



### Krok 3: Zapojení.





### Pokyny pro zapojení:

| Název modulu    | Vývod modulu | Vývod ESP32 |
|-----------------|--------------|-------------|
| Bílý LED modul  | S            | pin5        |
|                 | V            | 5V          |
|                 | G            | GND         |
| Dotykový senzor | S            | pin32       |
|                 | V            | 5V          |
|                 | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

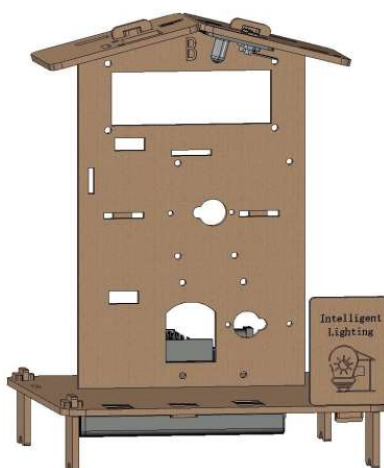
[Kliknutím sem otevřete program pro tento projekt.](#)



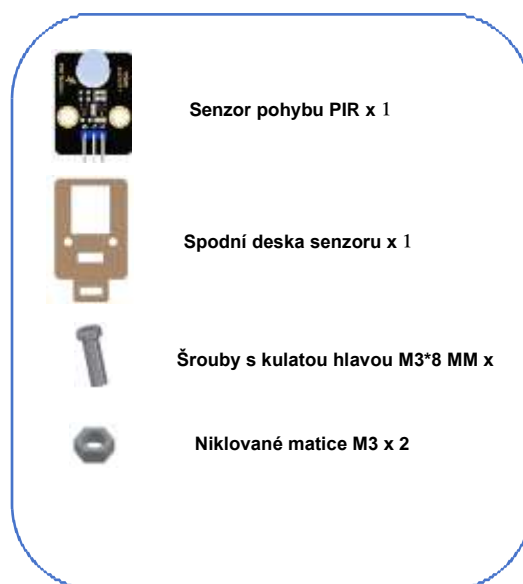
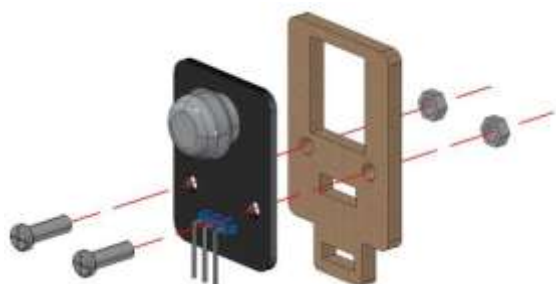


## 05 Inteligentní světlo

**Žhavá výzva: Struktura instalace LED v tomto projektu se bude řídit strukturou projektu 1 (zobrazena níže). Konkrétní kroky konstrukce zde nebudou uvedeny. Začneme instalací pohybového čidla PIR.**



**Krok 1: Instalace pohybového čidla PIR.**



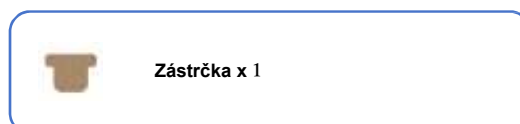


Krok 2: Instalace světelného senzoru.



Poznámka : Světelný senzor nainstalujte na dřevěnou desku D1 foto rezistorovou stranou nahoru.

Krok 3: Upevnění pohybového čidla PIR na základnu.





Krok 4: Oprava systému LUMI.



Dřevěná deska "C" x 1



Zástrčka

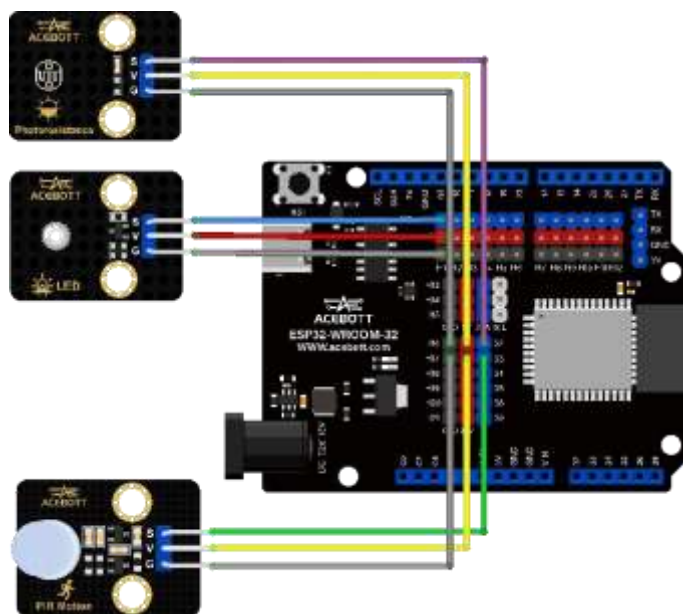


LIMI x 1





## Krok 5: Zapojení.



### Pokyny pro zapojení:

| Název modulu      | Vývod modulu | Vývod ESP32 |
|-------------------|--------------|-------------|
| Bílý LED modul    | S            | pin5        |
|                   | V            | 5V          |
|                   | G            | GND         |
| Světelný senzor   | S            | pin32       |
|                   | V            | 5V          |
|                   | G            | GND         |
| Senzor pohybu PIR | S            | pin33       |
|                   | V            | 5V          |
|                   | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)





## 06 RGB světlo

**Žhavá výzva: Základní struktura tohoto projektu je stejná jako u Projektu 1 (zobrazeného níže), Konkrétní kroky konstrukce zde nebudou uvedeny. Začneme instalací modulu RGB LED.**



Krok 1: Instalace modulu RGB LED.



RGB LED modul x 1



Niklované matice M3 x 2



Šrouby s kulatou hlavou M3\*8MM x

**Poznámka: Modul RGB LED nainstalujte pod dřevěnou desku D2 pomocí strana s kuličkami lampy směřuje dolů.**



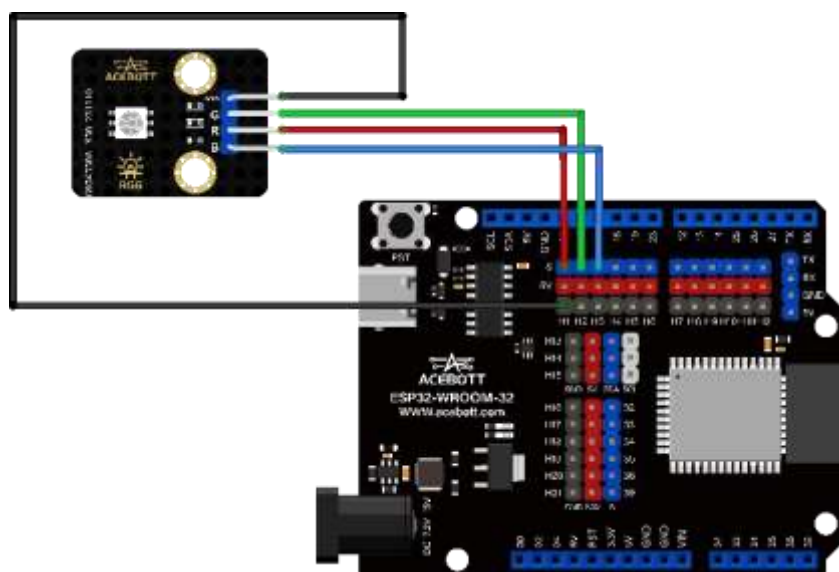


Krok 2: Upevnění stojící karty na základnu.





### Krok 3: Zapojení.



### Pokyny pro zapojení:

| Název modulu  | Vývod modulu | Vývod ESP32 |
|---------------|--------------|-------------|
| Modul RGB LED | GND          | GND         |
|               | G            | pin16       |
|               | R            | pin6        |
|               | B            | pin17       |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)





## 07 Elektronické kostky

**Žhavá výzva: Základní struktura tohoto projektu je stejná jako u projektu 1 (zobrazeného níže), Konkrétní konstrukční kroky zde nebudou uvedeny.**



Krok 1: Instalace dřevěné desky C.





Všimněte si: písmeny nahoru.

Krok 2: Instalace čtyřmístného modulu trubcového displeje.

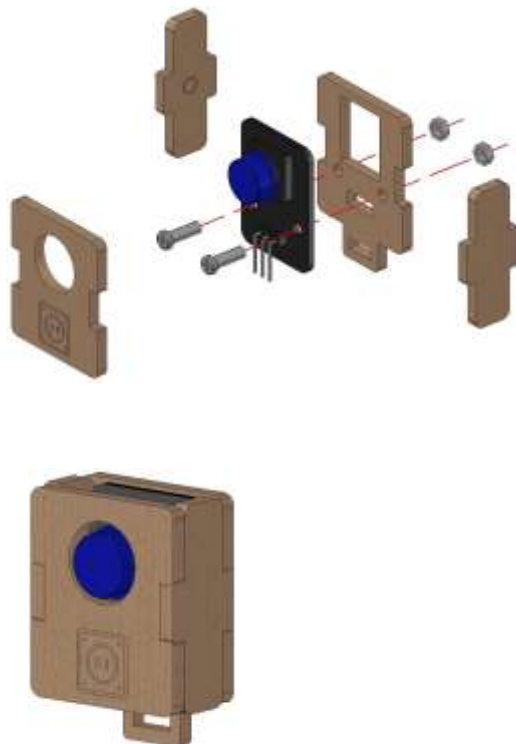


Poznámka: Nejprve připevněte modul 4místného trubcového displeje na spodní desku a oddělte určitý prostor pomocí těsnění mezi modulem 4místného trubcového displeje a spodní deskou. Poté nainstalujte boční desku, boční desku otvorem na straně kolíku digitální trubice a nakonec nainstalujte kryt.





Krok 3: Instalace tlačítkového modulu.



Tlačítkový modul x



Spodní deska knoflíkové



Kryt knoflíkové



Boční deska knoflíkové



Šrouby s kulatou hlavou M3\*8 MM x



Niklované matice M3 x 2

Krok 4: Upevnění modulu digitální trubice na plošinu a upevnění tlačítkového modulu na základnu.



Zástrčka x 2



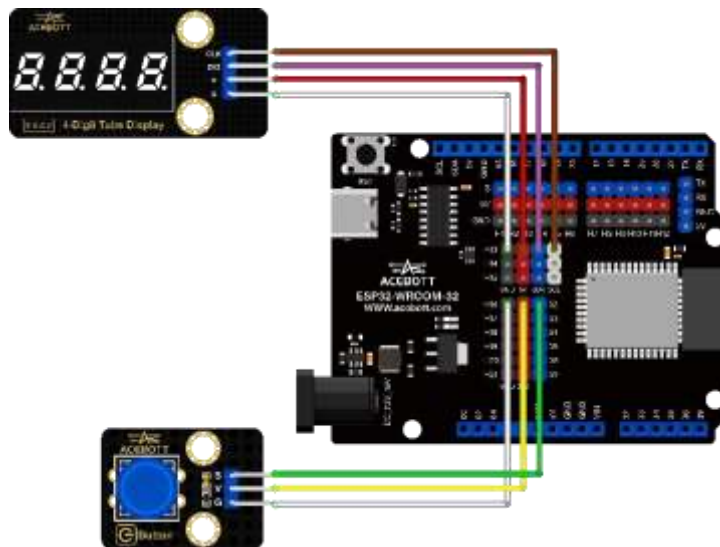


Krok 5: Upevnění stojící karty na základnu.





## Krok 6: Zapojení.



### Pokyny pro zapojení:

| Název modulu                        | Vývod modulu | Vývod ESP32 |
|-------------------------------------|--------------|-------------|
| Modul 4místného trubcového displeje | CLK          | SCL         |
|                                     | DIO          | SDA         |
|                                     | V            | 5V          |
|                                     | G            | GND         |
| Tlačítkový modul                    | S            | pin32       |
|                                     | V            | 3.3V        |
|                                     | G            | GND         |

**Poznámka:** Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

1. Před psaním programu je třeba nainstalovat soubor knihovny pro modul 4místného trubcového displeje. [Kliknutím sem získáte soubor knihovny "TM1650" pro 4místný trubcový displej.](#)

2. [Kliknutím sem otevřete program pro tento projekt.](#)

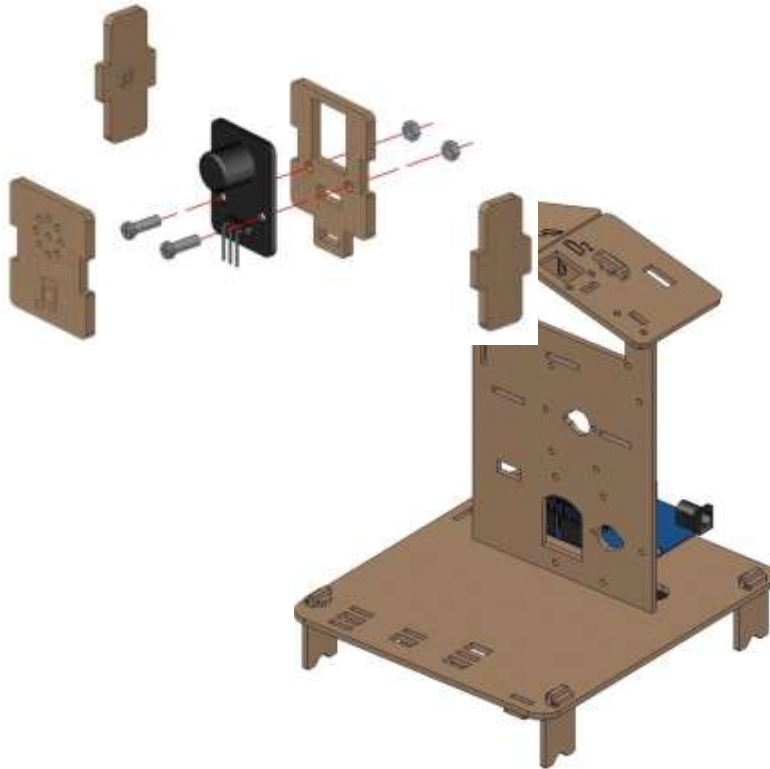




## 08 Elektronický bzučák

**Žhavá výzva: Začneme instalací modulu P-Buzzer.**

Krok 1: Instalace modulu P-Buzzer.



Modul P-Buzzer x 1



Spodní deska modulu P-Buzzer x 1



Kryt modulu P-Buzzer x 1



Boční deska modulu P-Buzzer x 2



Šrouby s kulatou hlavou M3\*8 MM x



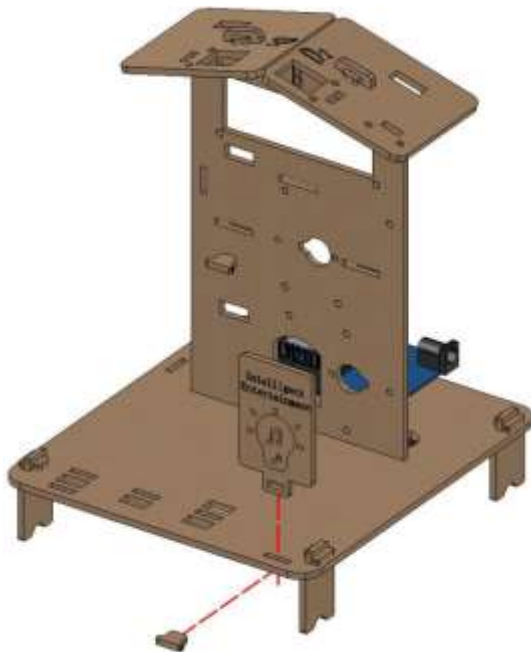
Niklované matice M3 x 2

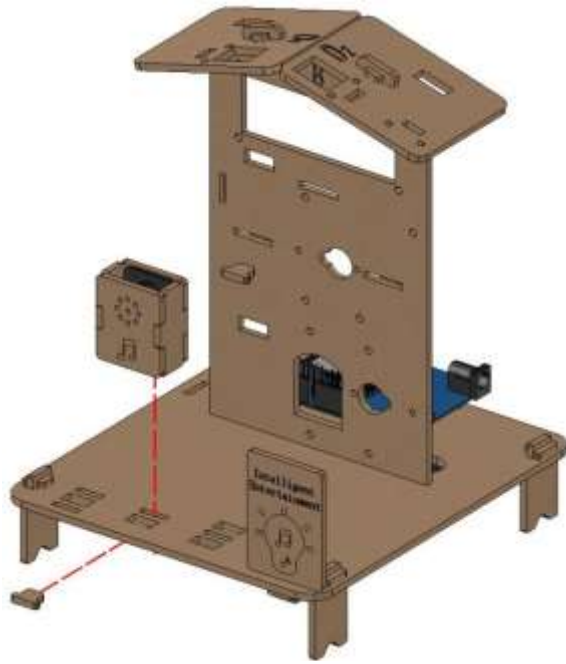




Poznámka: Nejprve připevněte modul P-Buzzer na spodní desku, poté nainstalujte boční desku , nakonec nainstalujte kryt.

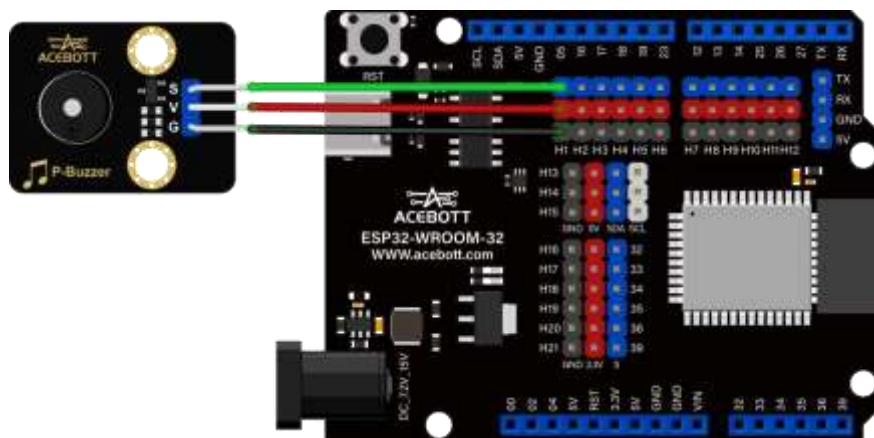
Krok 2: Upevnění stojící karty a modulu P-Buzzer.







### Krok 3: Zapojení.



### Pokyny pro zapojení:

| Název modulu   | Vývod modulu | Vývod ESP32 |
|----------------|--------------|-------------|
| Modul P-Buzzer | S            | pin32       |
|                | V            | 5V          |
|                | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)





## 09 Inteligentní přehrávač hudby

**Žhavá výzva: Začneme instalací modulu MP3.**

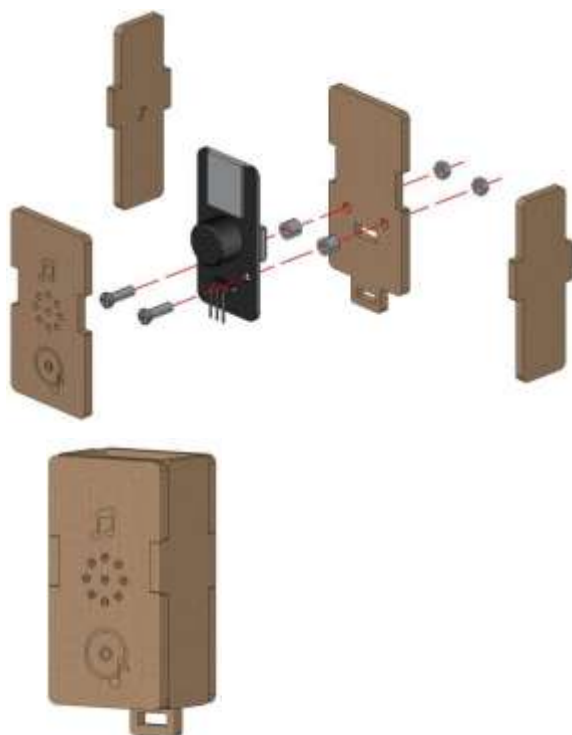


Poznámka: Před instalací modulu MP3 je třeba podle pokynů v učebnici stáhnout skladby na kartu SD modulu MP3.



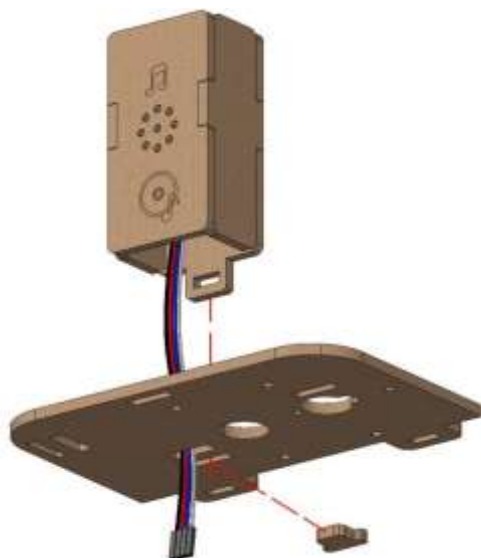


### Krok 1: Instalace modulu MP3.



Poznámka: Poté nainstalujte boční desku a nakonec nainstalujte kryt. 2. Upevněte modul Mp3 na spodní desku a oddělte určitý prostor s těsněním mezi modulem MP3 a spodní deskou.

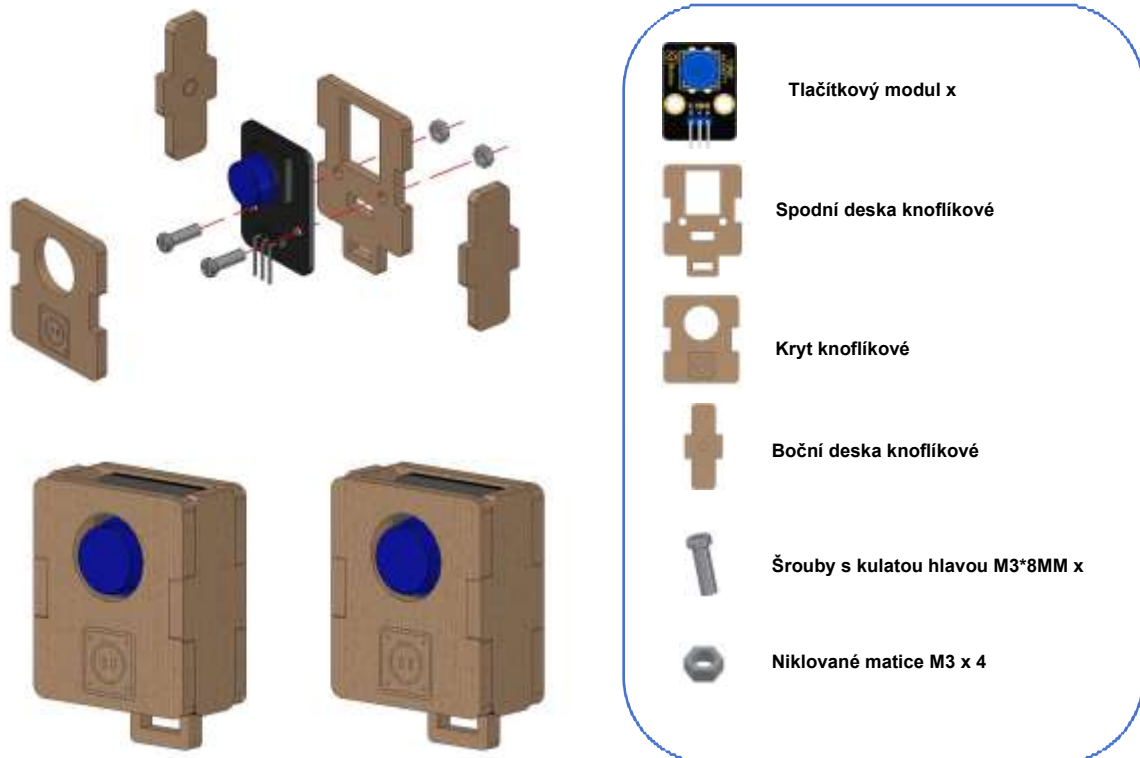
### Krok 2: Upevnění modulu MP3 na dřevěnou desku "C".



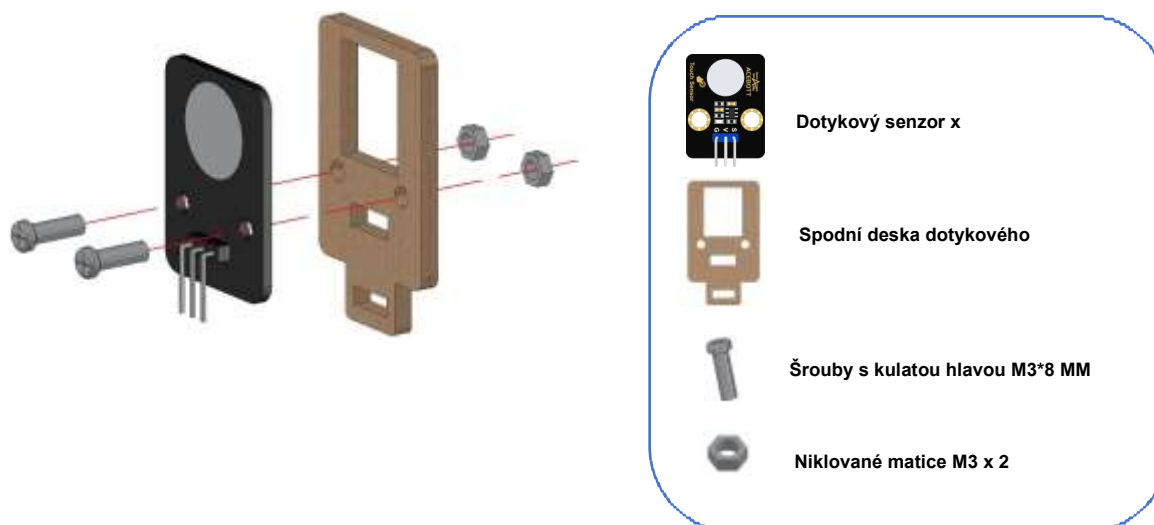


Poznámka: Umístěte desku "C" na okraj dřeva stranou, na které je písmeno, nahoru. Poté použijte čtyři jedno kolíkové vodiče DuPont typu samice-samice pro připojení ke kolíkům na přehrávači MP3. Nezapomeňte, kterému kolíku na přehrávači MP3 odpovídá který vodič DuPont. Poté připevněte modul MP3 přehrávače na desku "C" z basového dřeva.

Krok 3: Instalace dvou tlačítkových modulů.

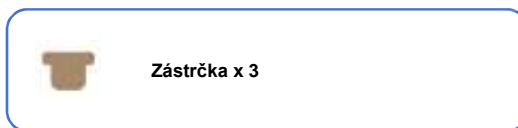
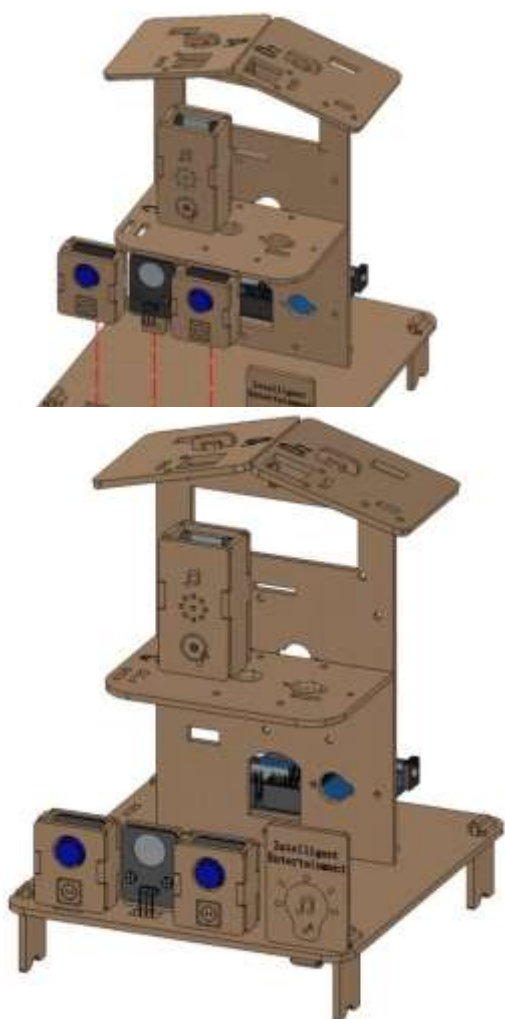


Krok 4: Instalace dotykového senzoru.



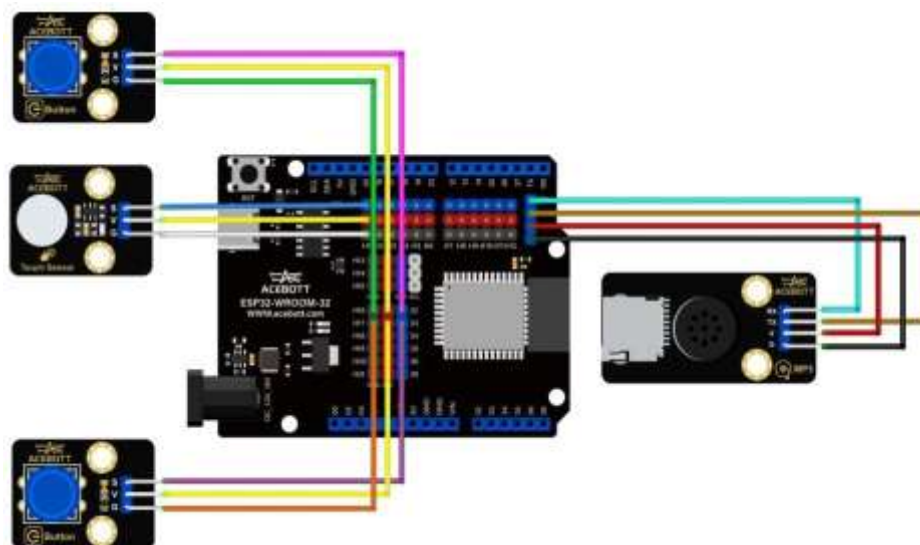


Krok 5: Upevnění tlačítkového modulu a dotykového senzoru na základnu.





## Krok 6: Zapojení.



### Pokyny pro zapojení:

| Název modulu      | Vývod modulu | Vývod ESP32 |
|-------------------|--------------|-------------|
| Modul MP3         | TX           | RX          |
|                   | RX           | TX          |
|                   | V            | 5V          |
|                   | G            | GND         |
| Tlačítkový modul1 | S            | pin32       |
|                   | V            | 3.3V        |
|                   | G            | GND         |
| Tlačítkový modul2 | S            | pin33       |
|                   | V            | 3.3V        |
|                   | G            | GND         |
| Dotykový senzor   | S            | pin5        |
|                   | V            | 5V          |
|                   | G            | GND         |

#### Poznámka:

1. Před nahráním programu odpojte vodiče Dupont mezi ESP32 a modulem MP3. Po úspěšném nahrání programu znovu připojte modul MP3 k ESP32 a poté stiskněte tlačítko RESET na ESP32, čímž program znovu spustíte.

2. Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.





1. Před psaním programu je třeba nainstalovat soubor knihovny pro modul MP3. [Kliknutím sem získáte soubor knihovny "Acebott" pro modul MP3.](#)

[Kliknutím sem otevřete program pro tento projekt.](#)





## 10 barevná světla

**Žhavá výzva : Základní struktura tohoto projektu je stejná jako v předchozím případě (viz níže), konkrétní kroky konstrukce zde nebudou uvedeny. Začneme instalací RGB LED pásku 10 LED.**



Krok 1: Instalace pásku RGB LED.





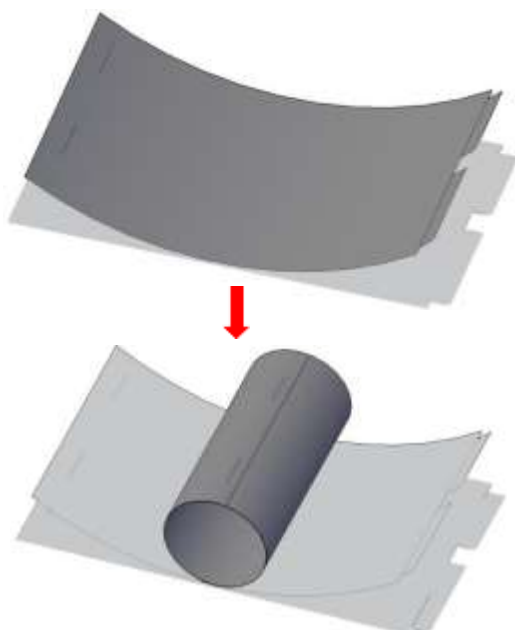
Poznámka: Po instalaci držáku RGB pásku strhněte fólii na zadní straně RGB LED pásku, poté omotejte RGB LED pásek kolem držáku a pevně jej rukama přitiskněte, aby zadní strana RGB LED pásku pevně přilnula k držáku. Při navíjení dbejte na to, aby jedna strana drátu pásku RGB LED byla dole.

Krok 2: Upevnění pásku RGB LED na plošinu.



Zástrčka x 1

Krok 3: Vyrobté stínítko.



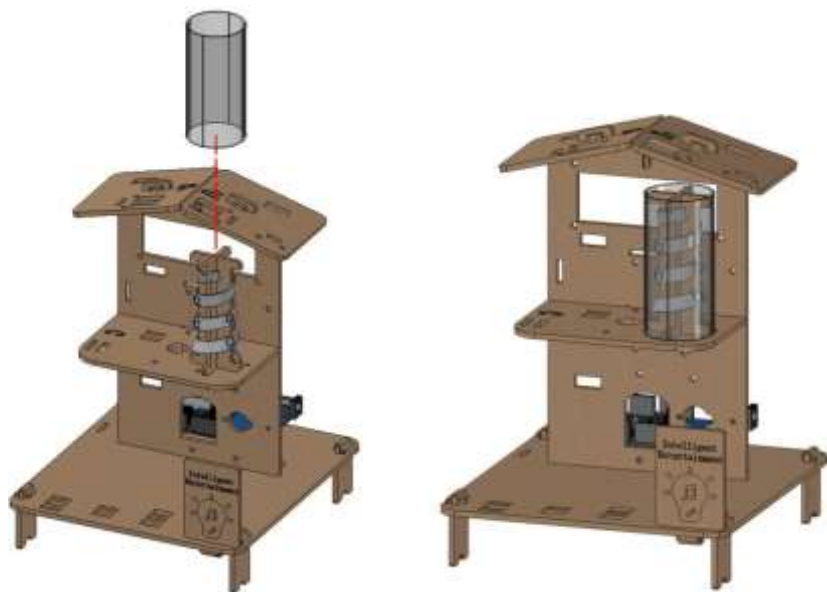
PVC materiál x 1

Poznámka: Stínidlo: PVC kartu srolujte do tvaru válce a na konci ji zajistěte sponou.

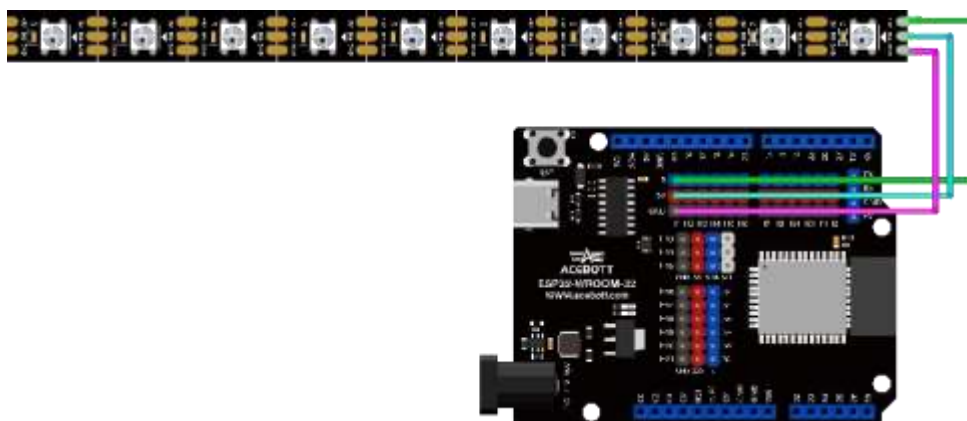




Krok 4: Nasadte stínítko na vnější stranu pásku RGB LED.



Krok 5: Zapojení.



**Pokyny pro zapojení:**

| Název modulu         | Vývod modulu | Vývod ESP32 |
|----------------------|--------------|-------------|
| RGB LED pásek 10 LED | Bílá čára    | pin5        |
|                      | Červená čára | 5V          |
|                      | Černá čára   | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

1. Před psaním programu je třeba nainstalovat soubor knihovny pro modul RGB Strip . [Klikněte na zde pro získání souboru knihovny](#).





["Adafruit NeoPixel" pro modul RGB Strip.](#)

2. Kliknutím sem otevřete program pro tento projekt.





## 11 Hudební světlo

**Žhavá výzva: Tento projekt bude i nadále používat strukturu RGB LED pásků (zobrazenou níže). Konkrétní kroky konstrukce zde nebudou uvedeny, a začneme instalací modulu MP3.**

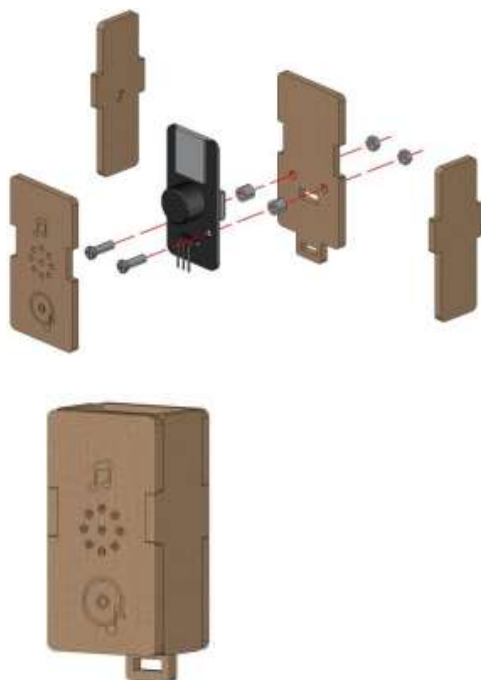


Poznámka: Před instalací přehrávače MP3 importujte skladby, které chcete přehrávat, do paměti MP3 podle pokynů v návodu.





### Krok 1: Instalace modulu MP3.



Modul MP3 x 1



Těsnění M3\*3 x 2



Spodní deska modulu MP3 x 1



Kryt modulu MP3 x 1



Boční deska modulu MP3 x

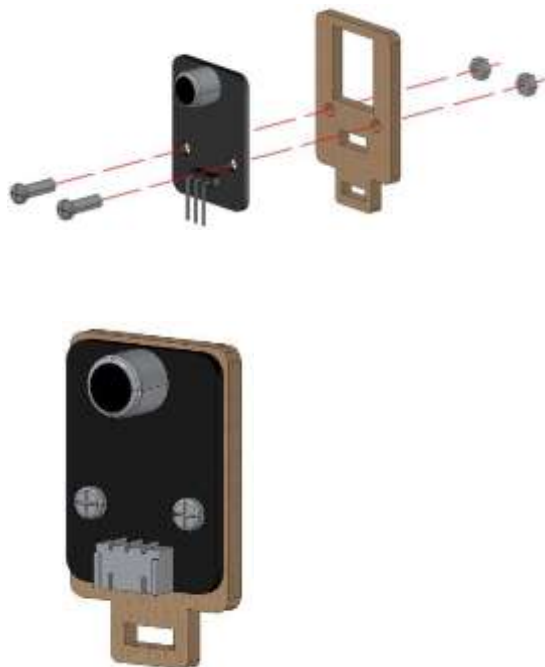


Niklované matice M3 x 2



Šrouby s kulatou hlavou M3\*12MM x

### Krok 2: Instalace snímače zvuku.



Snímač zvuku x 1



Spodní deska zvukového



Šrouby s kulatou hlavou M3\*8 MM x



Niklované matice M3 x 2

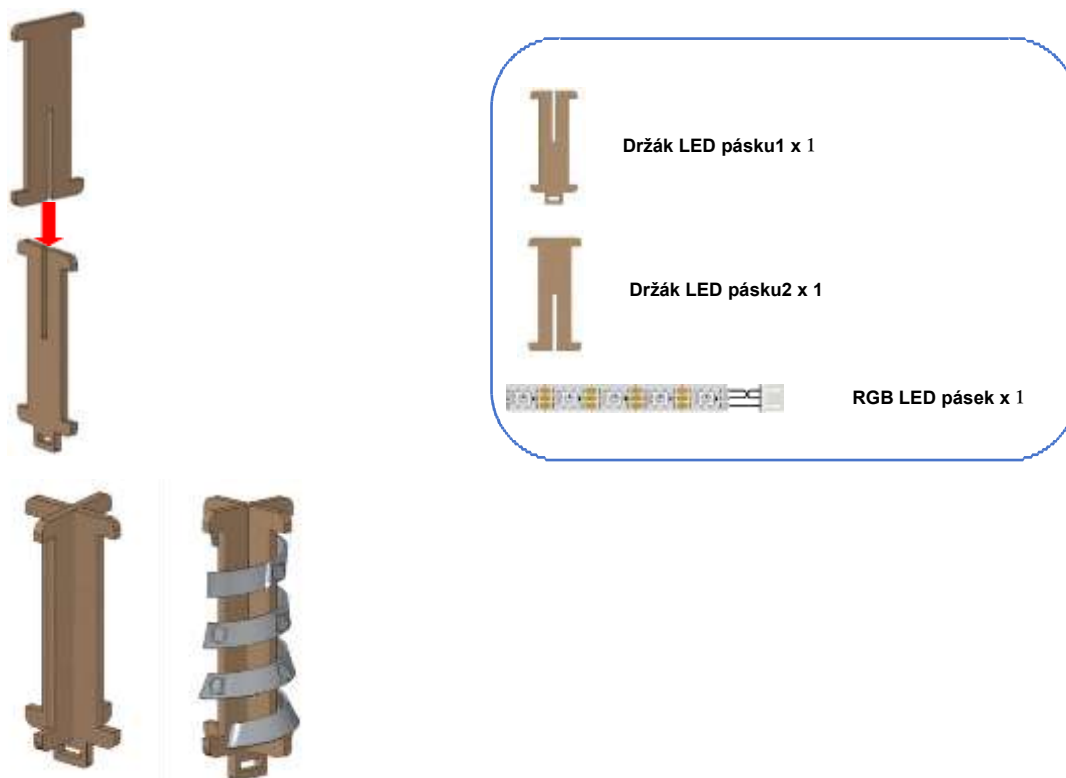


Zástrčka



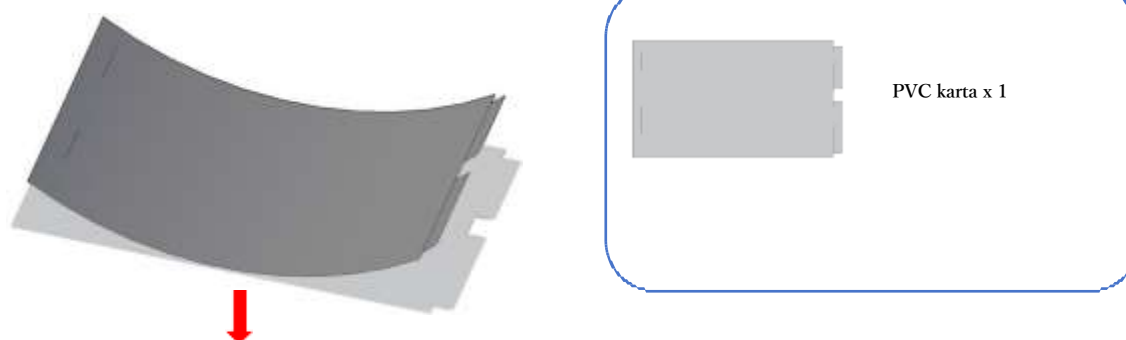


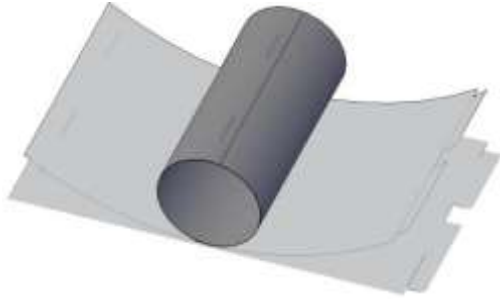
### Krok 3: Instalace pásku RGB LED.



Poznámka: Po instalaci držáku RGB pásku strhněte fólii na zadní straně RGB LED pásku, poté omotejte RGB LED pásek kolem držáku a pevně jej rukama přitiskněte, aby zadní strana RGB LED pásku pevně přilnula k držáku. Při navíjení dbejte na to, aby jedna strana drátu pásku RGB LED byla dole.

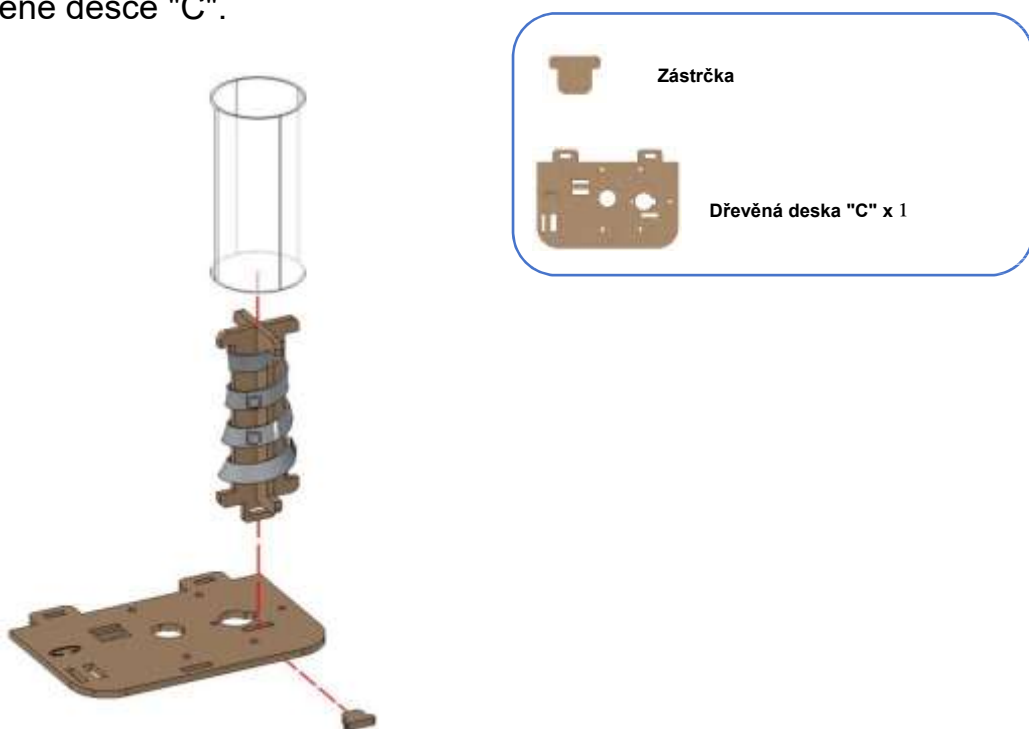
### Krok 4: Vyrobté stínítko.





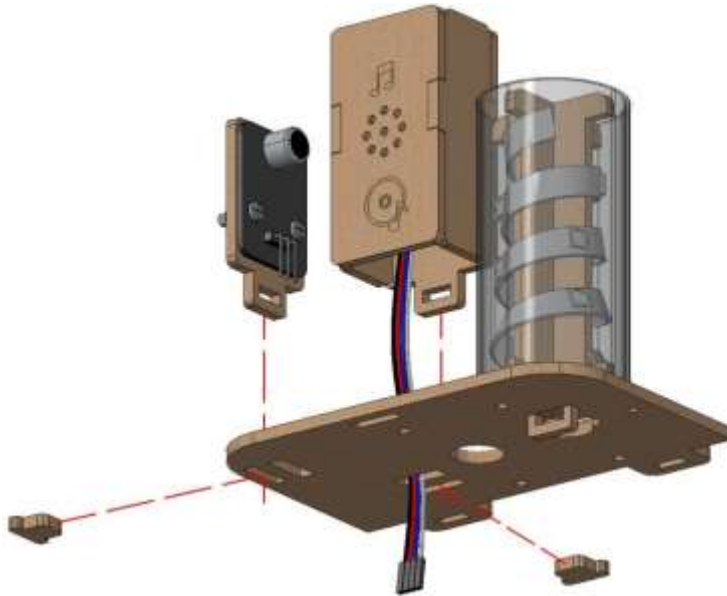
Poznámka: Stínidlo: PVC kartu srolujte do tvaru válce a na konci ji zajistěte sponou.

Krok 5: Připevnění pásky LED, zvukového senzoru a modulu MP3 k dřevěné desce "C".



Poznámka: Ujistěte se, že strana dřevěné desky s písmenem "A" směřuje nahoru.



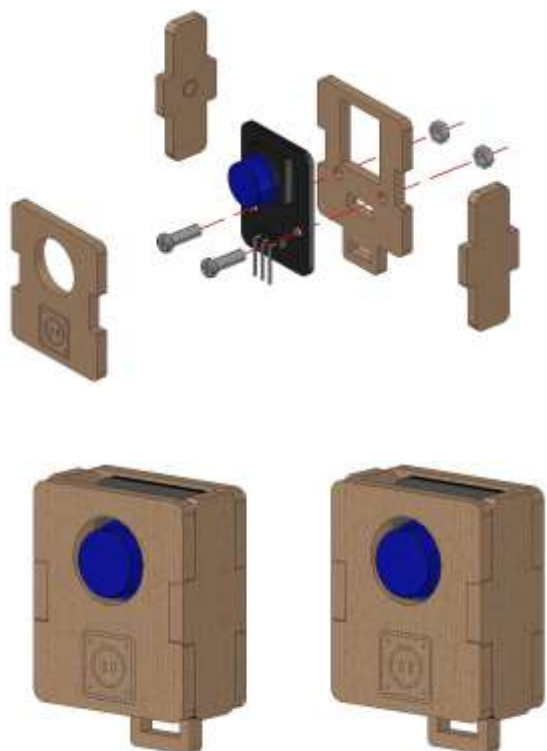


Poznámka: Pro připojení ke kolíkům na přehrávači MP3 použijte čtyři 1-pinové vodiče DuPont typu samice-samice. Pamatujte si, kterému kolíku na přehrávači MP3 odpovídá každý vodič DuPont. Poté upevněte modul MP3 přehrávače na desku "C" z basového dřeva.





### Krok 6: Instalace tlačítkového modulu.



Tlačítkový modul x



Spodní deska knoflíkové



Kryt knoflíkové



Boční deska knoflíkové

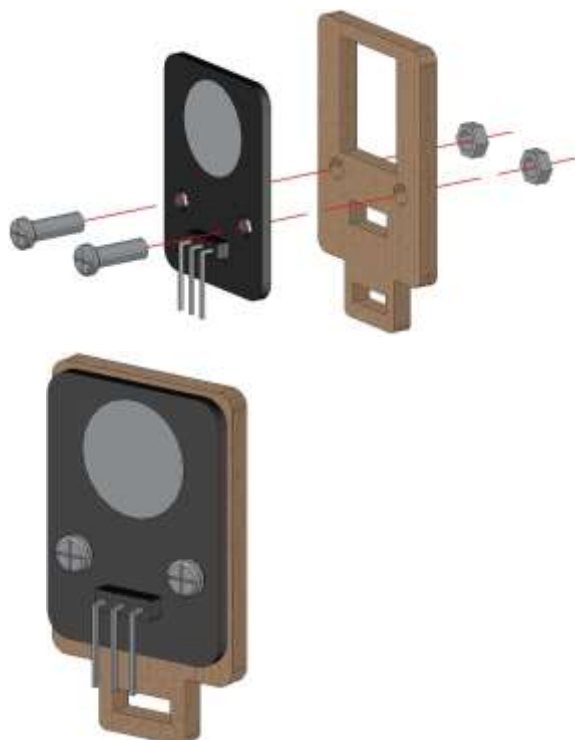


Šrouby s kulatou hlavou M3\*8MM x



Niklované matice M3 x 4

### Krok 7: Instalace dotykového senzoru.



Dotykový senzor x



Spodní deska dotykového



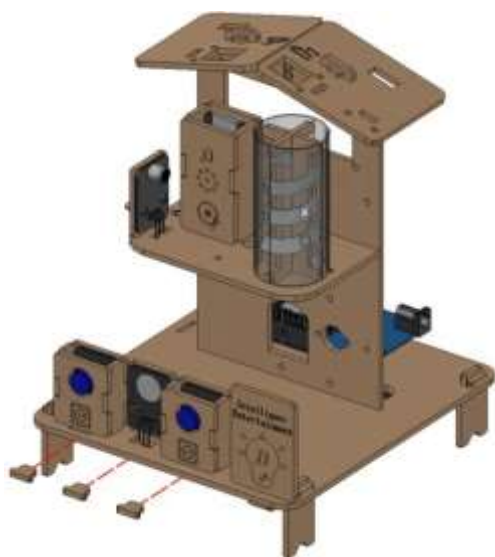
Šrouby s kulatou hlavou M3\*8 MM



Niklované matice M3 x 2

### Krok 8: Upevnění tlačítkového modulu a dotykového senzoru na desce základna.





Zástrčka x 3

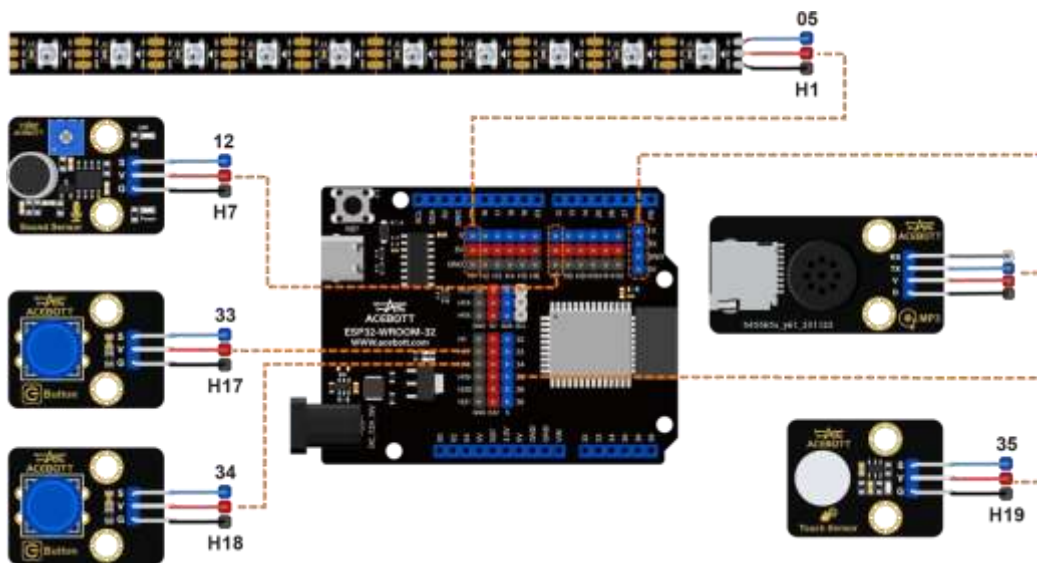


Krok 9: Oprava systému LUMI.





Krok 10: Zapojení.





## Pokyny pro zapojení:

| Název modulu         | Vývod modulu | Vývod ESP32 |
|----------------------|--------------|-------------|
| Modul MP3            | TX           | RX          |
|                      | RX           | TX          |
|                      | V            | 5V          |
|                      | G            | GND         |
| Tlačítkový modul1    | S            | pin33       |
|                      | V            | 3.3V        |
|                      | G            | GND         |
| Tlačítkový modul2    | S            | pin34       |
|                      | V            | 3.3V        |
|                      | G            | GND         |
| Dotykový senzor      | S            | Pin35       |
|                      | V            | 3.3V        |
|                      | G            | GND         |
| RGB LED pásek 10 LED | Bílá čára    | pin5        |
|                      | Červená čára | 5V          |
|                      | Černá čára   | GND         |
| Snímač zvuku         | S            | pin12       |
|                      | V            | 5V          |
|                      | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)





## 12 Inteligentní detekce T&H

**Žhavá výzva: Začneme instalací čtyřmístného modulu trubcového displeje.**

Krok 1: Instalace čtyřmístného modulu trubcového displeje.



Modul 4místného trubcového



Šrouby s kulatou hlavou M3\*12MMx



Niklované matice M3 x 2



Těsnění M3\*3 x 2



4místný modul trubcového displeje  
dole



Kryt modulu 4místného trubcového



4místný modul trubcového displeje levá boční



Modul 4místného trubcového displeje pravá boční



Zástrčka





Krok 2: Instalace čidla vlhkosti a teploty DHT11.



DHT11 Senzor vlhkosti a teploty x 1



Šrouby s kulatou hlavou M3\*8MM x



Niklované matice M3 x 2



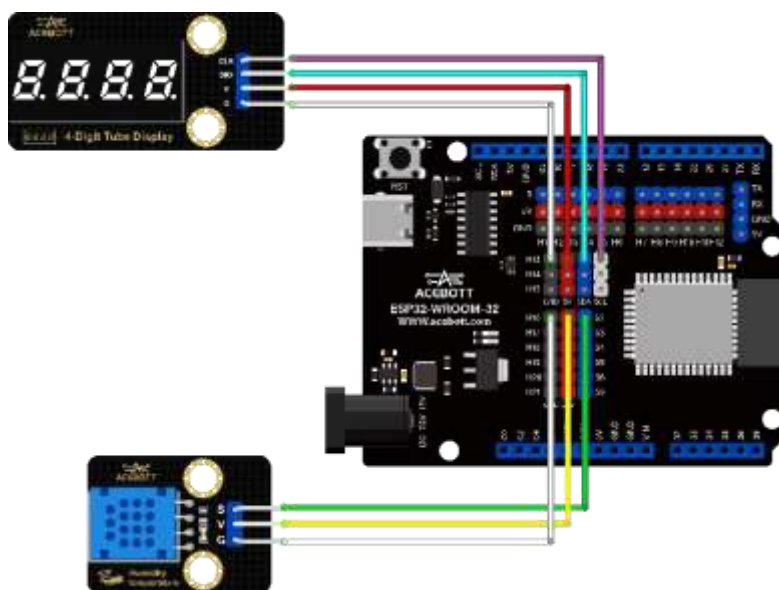


Krok 3: Upevnění stojící karty na základnu.





#### Krok 4: Zapojení.



#### Pokyny pro zapojení:

| Název modulu                        | Vývod modulu | Vývod ESP32 |
|-------------------------------------|--------------|-------------|
| Modul 4místného trubcového displeje | CLK          | SCL         |
|                                     | DIO          | SDA         |
|                                     | V            | 5V          |
|                                     | G            | GND         |
| DHT11 Snímač vlhkosti a teploty     | S            | pin32       |
|                                     | V            | 3.3V        |
|                                     | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

1. Před psaním programu je třeba nainstalovat soubor knihovny pro senzor DHT. [Kliknutím sem získáte soubor knihovny "DHT sensor library" pro senzor DHT.](#)

2. [Kliknutím sem otevřete program pro tento projekt.](#)



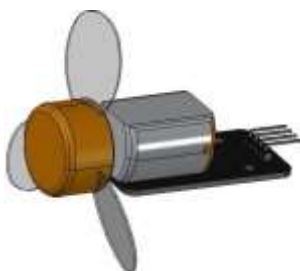
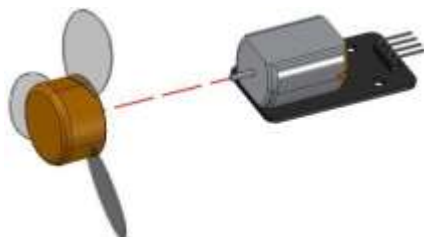


## 13 Inteligentní ventilátor

**Žhavá výzva: Tento projekt se bude řídit strukturou inteligentní detekce teploty a vlhkosti jako předchozí projekt (viz níže). Konkrétní kroky konstrukce zde nebudou uvedeny, začneme instalací modulu ventilátoru.**



Krok 1: Instalace lopatek ventilátoru.



130 Modul stejnosměrného



Lopatka



Šrouby s kulatou hlavou M3\*8MM x



Niklované matice M3 x 2



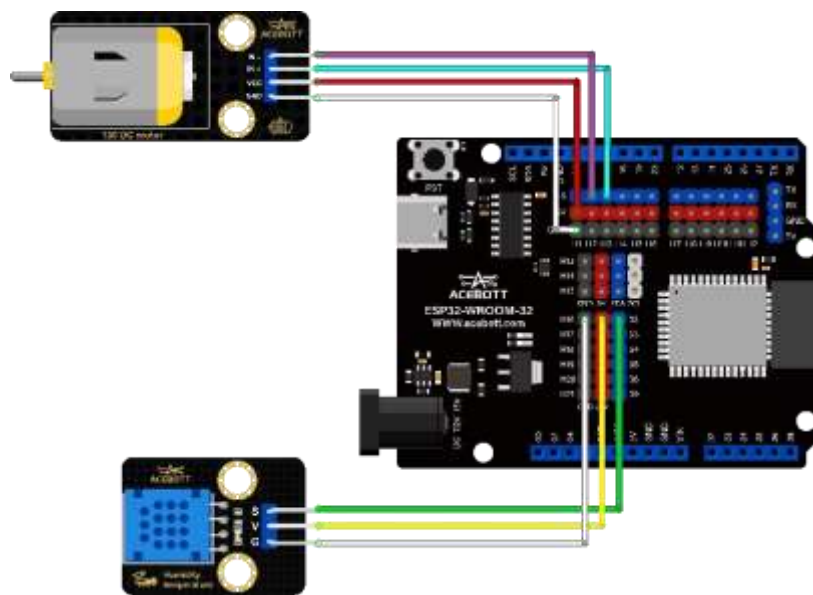


Poznámka: Nejprve nasadíte lopatku ventilátoru na hřídel motoru a poté celý modul ventilátoru připevníte na horní stranu dřevěné desky D2.





## Krok 2: Zapojení.



### Pokyny pro zapojení:

| Název modulu                    | Vývod modulu | Vývod ESP32 |
|---------------------------------|--------------|-------------|
| 130 Modul stejnosměrného motoru | IN-          | pin16       |
|                                 | IN+          | Pin17       |
|                                 | V            | 5V          |
|                                 | G            | GND         |
| DHT11 Snímač vlhkosti a teploty | S            | pin32       |
|                                 | V            | 3.3V        |
|                                 | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)

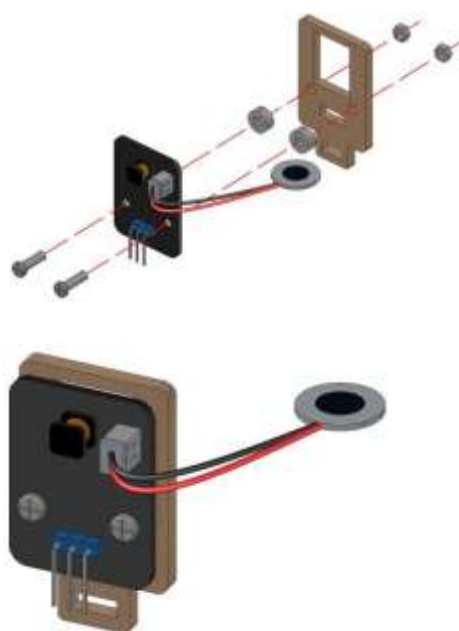




## 14 Inteligentní řídicí systém T&H

**Žhavá výzva: Tento projekt bude mít stejnou strukturu jako předchozí projekt (viz níže). Konkrétní kroky konstrukce zde nebudou uvedeny. Začneme instalací zvlhčovače Moudule.**

Krok 1: Instalace zvlhčovače Moudule.



Zvlhčovač vzduchu



Spodní deska senzoru x 1



Šrouby s kulatou hlavou M3\*12 MM x



Niklované matice M3 x 2

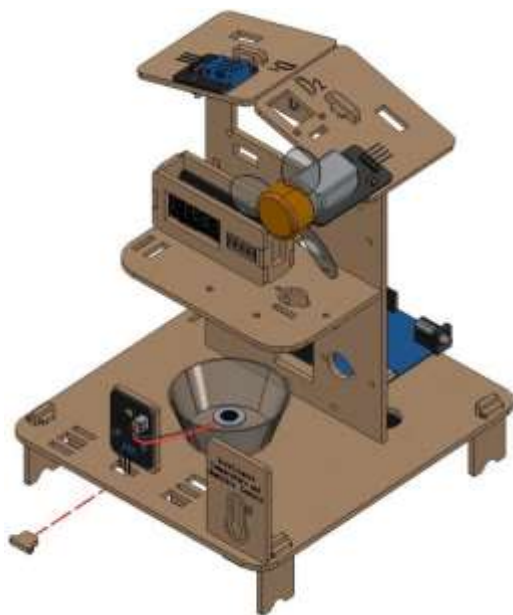


Těsnění M3\*3 x 2





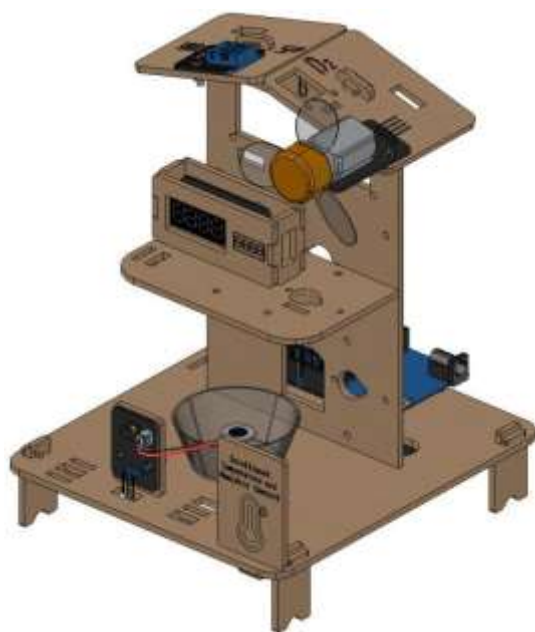
Krok 2: Upevnění zvlhčovače Moudule na základnu.



Plastová miska x



Zástrčka

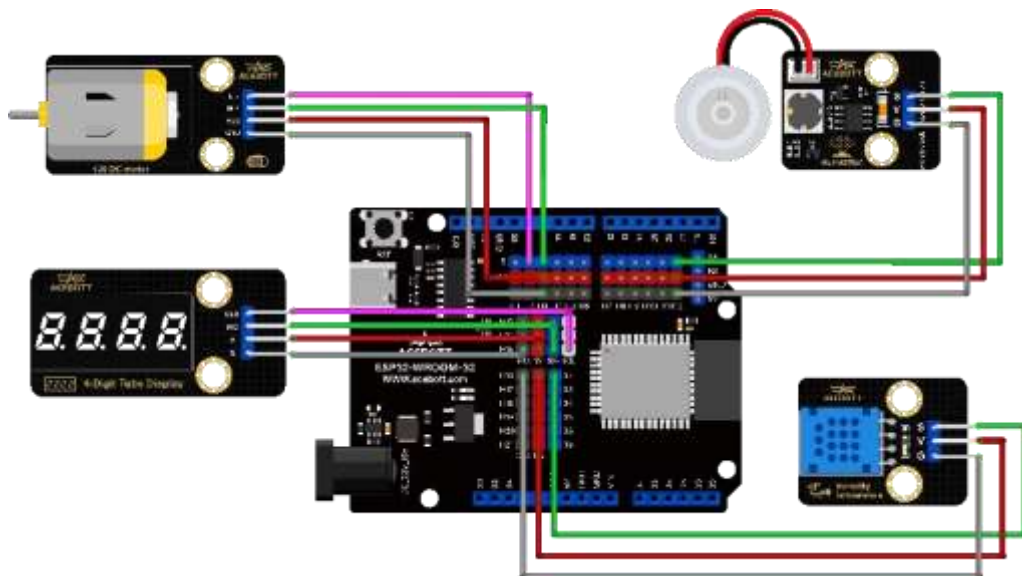


Krok 3: Upevnění LUMI.





#### Krok 4: Zapojení.





### Pokyny pro zapojení:

| Název modulu                        | Vývod modulu | Vývod ESP32 |
|-------------------------------------|--------------|-------------|
| Modul 4místného trubcového displeje | CLK          | SCL         |
|                                     | DIO          | SDA         |
|                                     | V            | 5V          |
|                                     | G            | GND         |
| DHT11 Snímač vlhkosti a teploty     | S            | pin32       |
|                                     | V            | 5V          |
|                                     | G            | GND         |
| 130 Modul stejnosměrného motoru     | IN-          | pin16       |
|                                     | IN+          | Pin17       |
|                                     | V            | 5V          |
|                                     | G            | GND         |
| Zvlhčovač vzduchu Moudule           | S            | Pin27       |
|                                     | V            | 5V          |
|                                     | G            | GND         |

Poznámka: Při připojování modulu k řídicí desce ESP32 se důsledně řiďte pokyny pro zapojení. Nesprávné zapojení může způsobit zkrat a poškodit řídicí desku ESP32.

[Kliknutím sem otevřete program pro tento projekt.](#)





Ahoj, jsem Lumi, robot z budoucnosti. Mým posláním je pomocí vědy a techniky změnit svět tak, aby byl lidský život pohodlnější, komfortnější a krásnější. Doufám, že se ke mně v této zábavné výzvě připojíte.

Začněme nejprve proměnou domácího života. V mých očích je domov nejteplejším přístavem a místem, kde si můžeme odpočinout a relaxovat v našem rušném životě. Abychom tento přístav učinili obyvatelnějším, můžeme k proměně domácího prostředí využít sílu technologií. Vydejme se na kouzelnou cestu, která změní náš život prostřednictvím technologie.





## Kapitola 1 Budoucí domov

V této kapitole se dozvíte, jak bude vypadat domácnost budoucnosti, a zjistíte, jak technologie Smart Home mění domácnost v interaktivní, efektivní a energeticky úsporný obytný prostor. Když se seznámíte s komponentami a aplikacemi systému Smart Home, budete si moci představit a navrhnout svůj vlastní domov budoucnosti.








## Oddíl 1 Chytrá domácnost



### Cíl učebního plánu

-  Porozumět konceptu, složení a použití inteligentní domácnosti
-  Pochopíte, jak funguje inteligentní domácnost
-  Prozkoumejte dopad inteligentní domácnosti na lidský život



Co by podle vás bylo třeba zlepšit nebo co vám v současném domácím životě nevyhovuje?



### Do It

Vyjmenujte části vašeho domova, které podle vás potřebují vylepšit, a pokuste se napsat nápady, jak problém vyřešit.

| Problém | Nápady na řešení |
|---------|------------------|
|         |                  |
|         |                  |
|         |                  |





V budoucnu budou všechny problémy, které jste uvedl, vyřešeny a my budeme žít ve vysoce inteligentních domech s mnoha pohodlnými a efektivními funkcemi. Představte si postel, která vás automaticky vzbudí, závěsy, které se automaticky roztáhnou v závislosti na slunečním světle, a hudbu, která se po probuzení tiše spustí. Tyto scény vypadají jako ze sci-fi filmů, v domácnosti budoucnosti je bude realizovat technologie Smart Home.

### Co je to chytrá domácnost?

Smart Home je systém pro řízení domácnosti. Pomocí technologie internetu věcí, technologie umělé inteligence, komunikační technologie a dalších pokročilých vědeckých a technických poznatků je tento systém schopen



technologickými prostředky a instalací řídicích systémů a inteligentních terminálů umožňuje dosáhnout komplexní automatizace, inteligence, pohodlí, bezpečnosti, úspory energie a integruje umělecké dekorace, které ukazují jedinečný půvab moderní vědy a techniky.



## Co umí chytrá domácnost?

### Inteligentní osvětlení

Osvětlovací systémy hrají v domácím prostředí zásadní roli. Ve srovnání s tradičním systémem osvětlení domácnosti může systém osvětlení inteligentní domácnosti dosáhnout inteligentnějších a energeticky úspornějších světelných efektů. Prostřednictvím použití



zvukem, dálkovým ovládáním, síťovou komunikací a dalšími způsoby ovládání, může účinně dálkově ovládat pracovní stav lampy, a dát tak sbohem tradičnímu způsobu přepínání tlačítka. Kromě toho má osvětlovací systém inteligentní domácnosti také funkci automatického rozpoznání okolního světla a automatického nastavení jasu, aby poskytoval pohodlnější, energeticky úspornější a humánnější světelné prostředí.

### Chytrá domácí zábava



Inteligentní systém domácí zábavy využívá umělou inteligenci a internetovou technologii k dosažení komplexní modernizace služeb domácí zábavy. Podle



náladu a preference uživatele, může systém automaticky doporučit vhodný filmový a televizní, hudební a herní obsah a automaticky upravit světelný efekt podle zábavného obsahu, aby vytvořil skutečnější a příjemnější atmosféru zábavy.

### **Inteligentní detekce prostředí**

Inteligentní systém detekce prostředí je schopen přesně detekovat různé faktory jako je teplota, vlhkost, kvalita vzduchu, kvalita vody a hluk v domácím prostředí. Pomocí chytrých zařízení systém



dokáže automaticky upravit prostředí do bezpečnějšího a pohodlnějšího stavu. Jako uvedme teplotu a vlhkost, které mohou díky přesné regulaci zajistit, aby byla v místnosti po celý rok udržována vhodná teplota a vlhkost, a zajistit tak uživatelům pohodlnější bydlení.

### **Inteligentní zabezpečení**

Inteligentní bezpečnostní systém pro rodinu, který poskytuje pokročilá a komplexní bezpečnostní opatření zajištění osobní bezpečnosti.





neporušené rodinné příslušníky a majetek. Systém má inteligentní funkce kontroly přístupu, alarmu proti vloupání, požárního alarmu a alarmu při živelných pohromách, které mohou účinně předcházet a reagovat na různé potenciální

bezpečnostní hrozby. Díky inteligentnímu bezpečnostnímu systému mohou členové rodiny využívat bezpečnější a pohodlnější životní prostředí.

### **Chytrý úklid**

Systém chytré domácnosti dokáže automaticky provádět domácí práce, jako je automatické zametání podlahy, automatické krmení domácích zvířat a automatické zalévání květin. Výrazně snižuje domácí práce, zátěž rodinných příslušníků, zvyšuje pohodlí a zlepšuje kvalitu života a spokojenost lidí.



## Jaké jsou součásti inteligentní domácnosti?

Když plánujeme vybudovat systém Smart Home, musíme nejprve znát jeho součásti. Systém Smart Home se skládá především z hardwaru a softwaru. Hardware jako "tělo" systému Smart Home se skládá z různých inteligentních zařízení a je zodpovědný za dosažení konečné funkce. Software je "vědomím" systému Smart Home, který řídí chytrá zařízení, aby plnila úkoly podle předem nastavených pokynů a realizovala různé inteligentní funkce.

Hardware inteligentní domácnosti se skládá z řídicí jednotky, senzoru, akčního členu a síťového komunikačního zařízení.



**Řídicí jednotka** se obvykle skládá z centrální procesorové jednotky (CPU), jednočipového mikropočítače (MCU) nebo inteligentní řídicí jednotky brány. Je to mozek inteligentní domácnosti

systém, který dokáže analyzovat a vypočítat všechny druhy shromážděných informací a učinit odpovídající rozhodnutí.

**Senzor** hraje v systému inteligentní domácnosti důležitou roli a je snímací jednotkou systému inteligentní domácnosti. Senzory jsou schopny shromažďovat různé informace o prostředí



informace, jako je světlo, teplota, vlhkost atd., čímž poskytuje



potřebná data zadávaná do systému.



**Akční členy** se týkají všech druhů domácích spotřebičů, které je třeba ovládat v inteligentní domácnosti, jako jsou lampy, klimatizace, televizory, systémy kontroly přístupu atd.

**Síťová komunikační zařízení** zahrnují kabelová komunikační zařízení a bezdrátová komunikační zařízení, která se používají především k realizaci datové komunikace mezi zařízeními. Umožňuje pohodlnější a efektivnější ovládání chytré domácnosti.



Softwarová část inteligentní domácnosti se dělí na vestavěný program a uživatelský ovládací program.



**Vestavěný program** je vyvinut na základě řídicí jednotky inteligentní domácnosti. Vývojáři si pro psaní vestavných programů zvolí odpovídající operační systém a programovací jazyk. Řídicí jednotka

realizuje odpovídající funkce každého inteligentního hardwaru prostřednictvím vestavěného programu.





**Uživatelský ovládací program** je vyvinut na základě jiných koncových zařízení, jako jsou mobilní telefony, které zahrnují podobu rozhraní dotykové obrazovky, smart phone APP, voice



asistenta atd. Jeho cílem je realizovat interakci mezi uživateli a celým systémem Smart Home a zlepšit pohodlí dálkového ovládání.



### Think About It

Chytrá domácnost nám přináší mnoho pohodlí, ale zároveň ji budou provázet i některé problémy. Zamyslete se nad výhodami a problémy, které chytrá domácnost přináší do našeho života, a napište svůj názor do níže uvedeného prostoru.

**Výhody chytré domácnosti:**

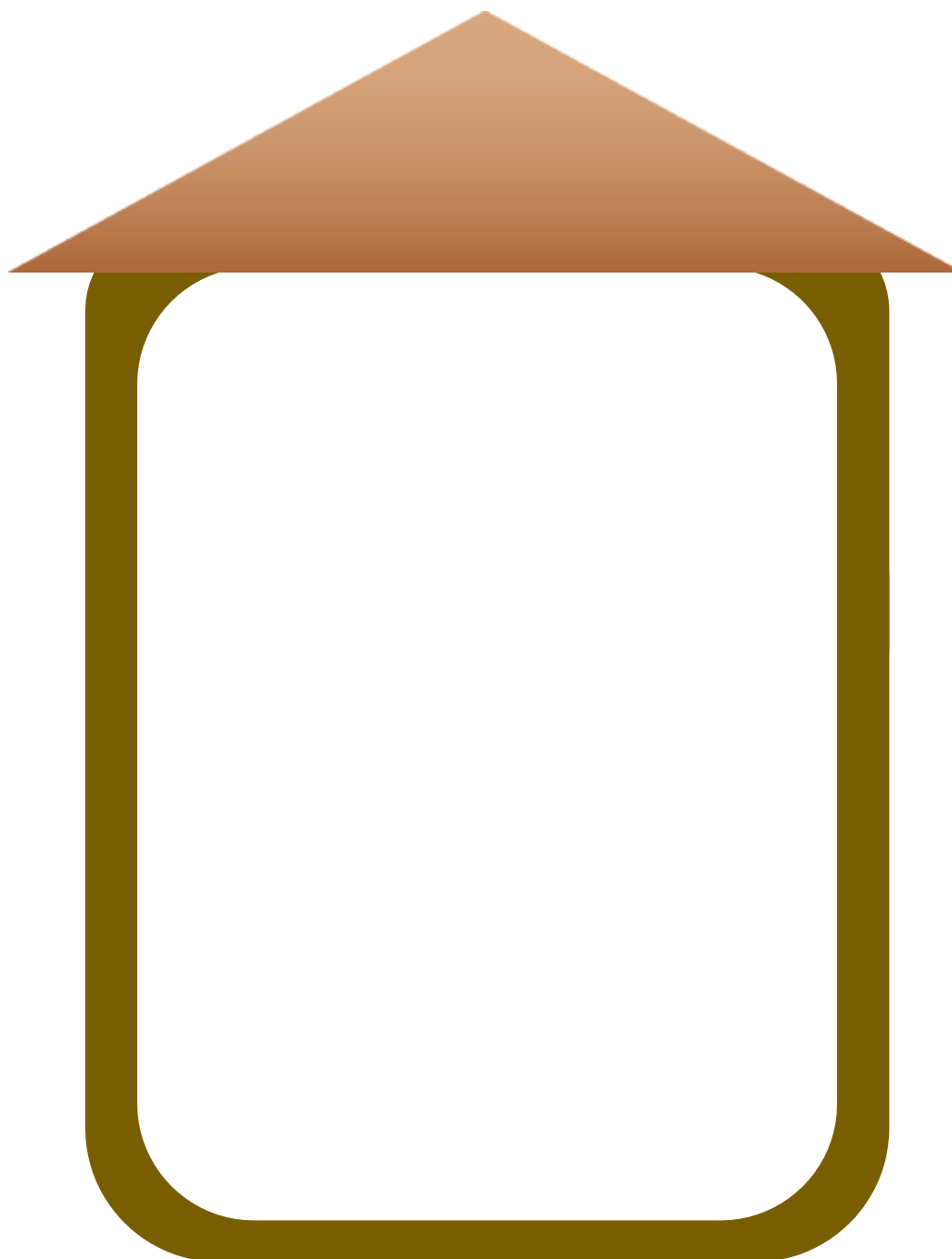
**problémy chytré domácnosti:**





**Do It**

Kdybyste si měli postavit vlastní chytrý dům, co byste v něm chtěli mít? Popište slovy nebo obrázky na níže uvedeném místě chytrý dům, který chcete vytvořit.



## ○ Oddíl 2 Mozek chytré domácnosti ○



### Cíl učebního plánu

- ✎ Porozumět konceptu jednočipového mikropočítače
- ✎ Porozumění struktuře a funkcím systému ESP32
- ✎ Zjistěte, co Arduino IDE dělá a co umí
- ✎ Zvládnutí způsobu instalace prostředí Arduino IDE
- ✎ Zvládnutí používání prostředí Arduino IDE
- ✎ Zkušenosti na proces na ESP32 programování a stahování



**Co je podle vás nejdůležitější součástí systému chytré domácnosti? A proč?**

Nejzásadnějším rozdílem mezi inteligentní a tradiční domácností je inteligence. Chytrá domácnost je chytrá, protože má mozek, řídicí jednotku. Centrální řídicí jednotkou inteligentní domácnosti může být počítač, jednočipový mikropočítač, krabička s bránou a další formy. Jeho hlavní funkcí je přijímat informace, poté je analyzovat a zpracovávat a vydávat odpovídající řídicí pokyny k realizaci automatizace a inteligence.

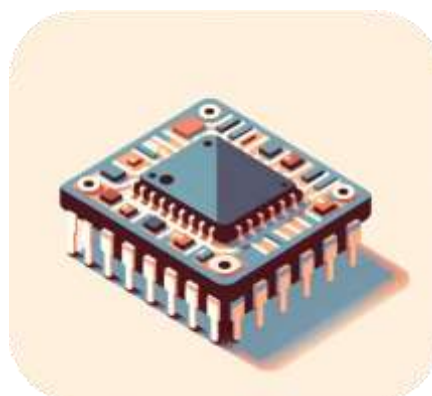


domácí vybavení.

V následujícím kurzu budeme používat MCU jako řídicí jednotku inteligentní domácnosti.

## Jednočipový mikropočítač

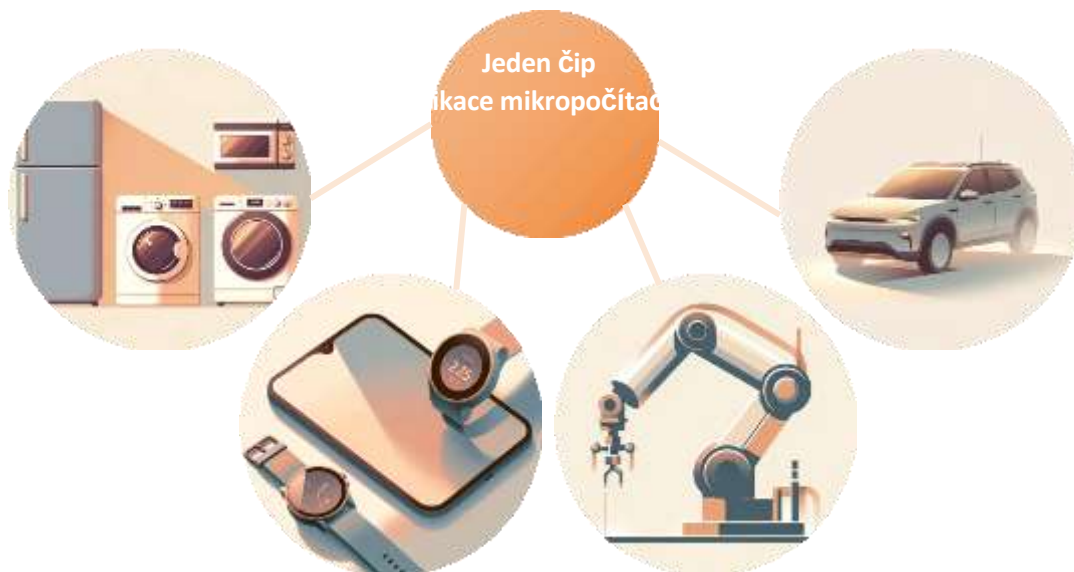
Jednočipový mikropočítač je čip s integrovaným obvodem, který integruje jádro mikroprocesoru s pamětí (např. RAM a ROM), vstupními/výstupními porty (I/O), časovači a dalšími funkcemi a tvoří tak kompletní mikropočítačový systém.



Mikro kontrolér je jako miniaturní počítač, který může pracovat samostatně bez externího podpůrného hardwaru. Díky této konstrukci je mikro kontrolér velmi vhodný pro vestavěné systémy a inteligentní zařízení a lze jej naprogramovat k provádění specifických úloh.

Mikro kontrolér je malý, úsporný, snadno se programuje, snadno se integruje a má další vlastnosti, takže se široce používá ve všech oblastech života, například v domácích spotřebičích, průmyslovém řízení, automobilové elektronice, lékařských přístrojích atd.





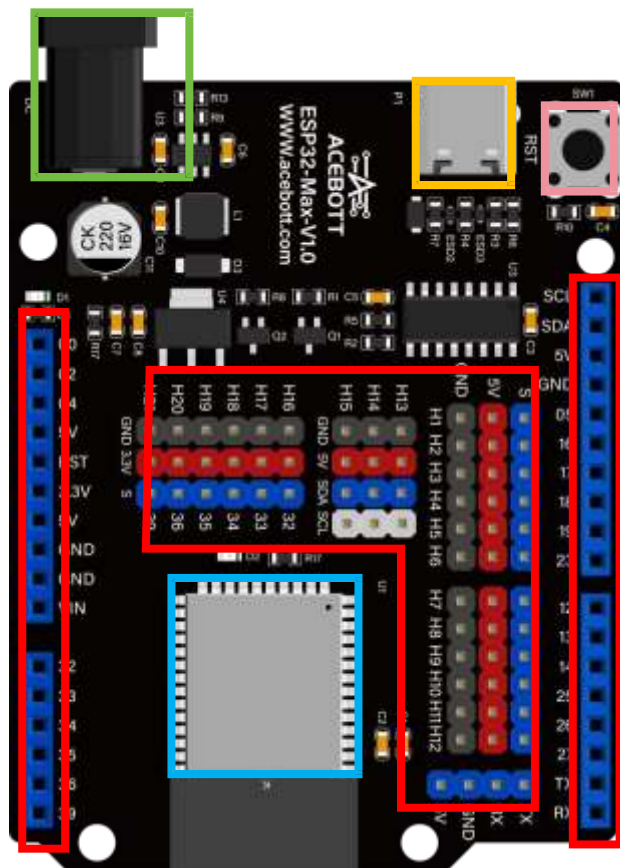
Existuje mnoho typů jednočipových mikročítačů, podle potřeby si můžeme vybrat vhodný model. Model jednočipového mikročítače použitý v tomto kurzu je ESP32.

### ESP32

ESP32 je vysoce integrovaný jednočipový mikročítač s nízkou spotřebou energie a duálním režimem Wi-Fi a Bluetooth vyvinutý společností Espressif Systems. Řídicí deska ESP32 je velmi oblíbená díky svým výkonným funkcím a bohatým komunikačním rozhraním, která jsou vhodná pro různé aplikace internetu věcí (IoT), vývoj vestavěných systémů a realizaci projektů.

Následuje schéma struktury řídicí desky Acebot-ESP32:





**Příkon:** Pro napájení řídicí desky lze připojit napájecí zdroj 7,2 V-15 V.

**Rozhraní Type-C:** Pro stahování programů, také může poskytovat 5V napájení pro řídicí desku.

**Klíč pro resetování:** Reset systému řídicí desky.

**GPIO:** Obecný vstup a výstup rozhraní pro vstupní a výstupní data.

**Moduly ESP32:** Obsahuje ESP32 mikroprocesor, Bluetooth, WiFi.

### Proč si vybrat ESP32?

ESP32 je velmi oblíbený, protože má následující vlastnosti:

- (1) Vysoký výkon: 240MHz dvoujádrový procesor, 520KB RAM a 4MB flash paměť.
  - (2) Nízká spotřeba energie: S řadou úsporných funkcí nejnižší spotřeba energie je pouze 5  $\mu$  A.
  - (3) Moduly WiFi a Bluetooth: Vestavěné moduly WiFi a Bluetooth
- 4.2 moduly pro bezdrátovou komunikaci.
- (4) GPIO: Díky 34 portům GPIO lze připojit a ovládat různé periferie.



(5) Zabezpečení: Podporuje bezpečnostní funkce, jako je šifrování a digitální podpisy.

Po získání řídicí desky ESP32 není její počáteční stav schopen implementovat žádné specifické funkce. Pokud chcete, aby měla konkrétní funkci, musíte ji naprogramovat. K tomu je třeba nejprve nakonfigurovat vývojové prostředí vhodné pro ESP32, které vývojářům poskytne programovací platformu. Mezi běžně používané vývojové platformy ESP32 patří MicroPython, Arduino IDE, ESP-IDF atd. Tento kurz se zaměří na použití Arduino IDE jako vývojové platformy ESP32.



## Výklad terminologie

### Vývojové prostředí

Vývojové prostředí, běžně známé také jako integrované vývojové prostředí (IDE), je aplikace, která poskytuje prostředí pro vývoj programu, obvykle zahrnující editor kódu, kompilátor, ladicí program a nástroje grafického uživatelského rozhraní. Integruje funkci psaní kódu, funkci analýzy, funkci kompilace, funkci ladění a další integrovanou sadu softwarových služeb.



Veškerý software nebo balíčky (skupiny) s touto funkcí lze nazvat integrovaným vývojovým prostředím.



## IDE Arduino

Arduino je pohodlná, flexibilní a snadno použitelná platforma pro tvorbu elektronických prototypů s otevřeným zdrojovým kódem, která zahrnuje hardware (různé modely řídicích desek s mikro kontroléry) a software (Arduino IDE). Při použití Arduino IDE stačí napsat program na ArduinoIDE, poté jej nahrát na řídicí desku příslušného modelu a řídicí deska bude pracovat podle pokynů programu.

Arduino IDE je bezplatný multiplatformní programovací nástroj pro psaní, nahrávání a ladění kódu Arduina. Kód Arduino je založen na programovacím jazyce C++ a poskytuje mnoho pokročilých knihoven a funkcí, které zjednodušují hardwarové

ovládání a interakce, což usnadňuje učení a používání začátečníkům.

Arduino IDE lze spustit na operačních systémech Windows, Mac a Linux. V této lekci se naučíme, jak nainstalovat Arduino IDE na Windows.



## Jak nainstalovat prostředí Arduino IDE v systému Windows?

Krok 1: Stáhněte si instalační program Arduino IDE z oficiálních webových stránek. Musíme přejít na oficiální webové stránky Arduina: <https://www.arduino.cc/en/Main/Software>. Podle následujících kroků stáhněte Arduino IDE a nainstalujte program.

1. Po otevření webové stránky vyberte odpovídající Arduino Verze IDE podle verze systému Windows vašeho počítače. Pokud je váš počítač ve verzi Win10 nebo vyšší, vyberte Arduino 2.x; pokud je váš počítač ve verzi před Win10, vyberte ke stažení verzi Arduino 1.x.

The screenshot displays the Arduino IDE download page. It features two main sections for different versions of the IDE. The top section is for Arduino IDE 2.2.1, and the bottom section is for Arduino IDE 1.8.19. Each section includes a description of the IDE's features and a 'DOWNLOAD OPTIONS' sidebar. In the 2.2.1 section, the 'Windows Win 10 and newer, 64 bits' option is highlighted with a red box. In the 1.8.19 section, the 'Windows Win 7 and newer' option is highlighted with a red box. The 'DOWNLOAD OPTIONS' sidebar for each version lists download links for Windows, Linux, and macOS, along with release notes and checksums.

**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.

**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

**Arduino IDE 1.8.19**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for installation instructions.

**SOURCE CODE**

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

**DOWNLOAD OPTIONS**

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 [Get](#)
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

Release Notes

Checksums (sha512)



## Tips

Posouváním stránky naleznete různé verze IDE Arduino.

**1** Download Arduino IDE & support it's progress  
Since the release 1.a release in March 2015, the Arduino IDE has been downloaded **77,922,677** times — impressive! Help its development with a donation.

\$3 \$5 \$10 \$25 \$50 Other

CONTRIBUTE AND DOWNLOAD  
OR  
JUST DOWNLOAD

Stay in the Loop: Join Our Newsletter! **2**  
As a beginner or advanced user, you can find inspiring projects and learn about cutting-edge Arduino products through our **weekly newsletter!**

email \*

I confirm to have read the [Privacy Policy](#) and to accept the [Terms of Service](#) \*

I would like to receive emails about special deals and commercial offers from Arduino.

SUBSCRIBE & DOWNLOAD  
OR  
JUST DOWNLOAD

**3**

Thank you for downloading!  
You're always welcome to sign up in the future. Keep up with exclusive updates, tutorials, events and join our community to empower creators worldwide anytime.

GO TO HOME PAGE

**4** arduino-ide\_2.2.1\_Windows\_64bit.exe

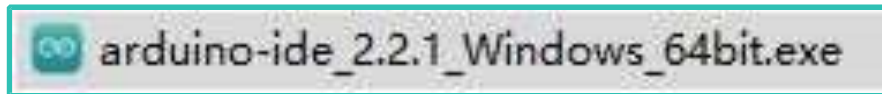
Krok 2: Dvakrát klikněte na stažený instalační program a podle níže uvedených pokynů nainstalujte Arduino IDE.



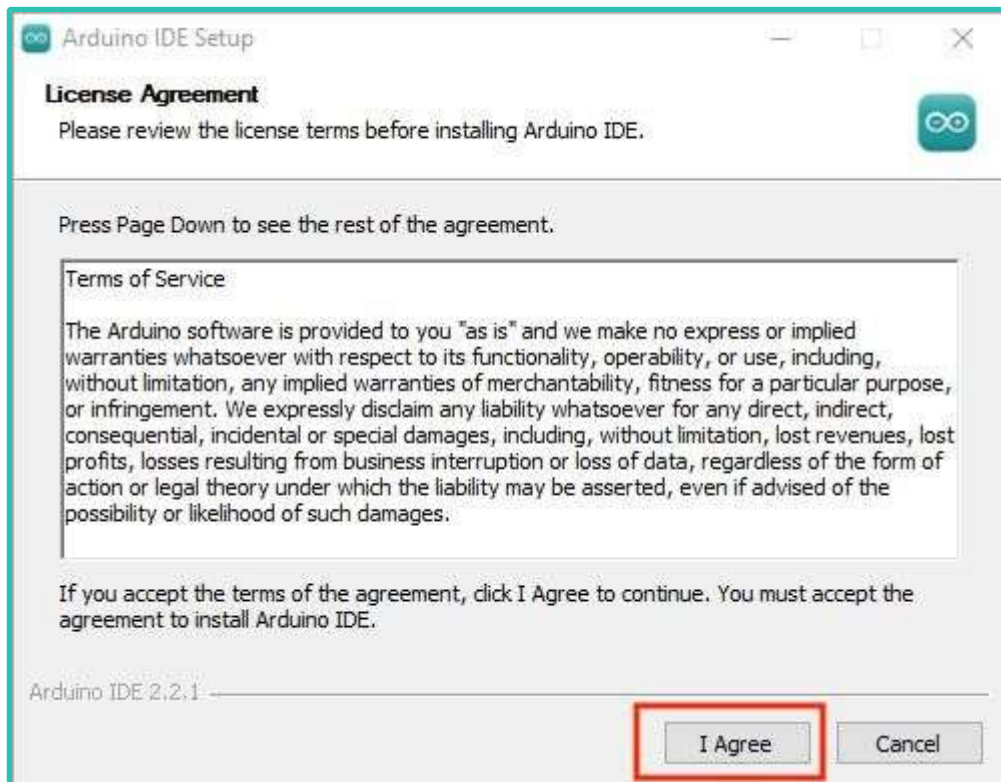


1. Po dokončení stahování se zobrazí soubor s ikonou.

Klikněte na tlačítko Instalovat software.

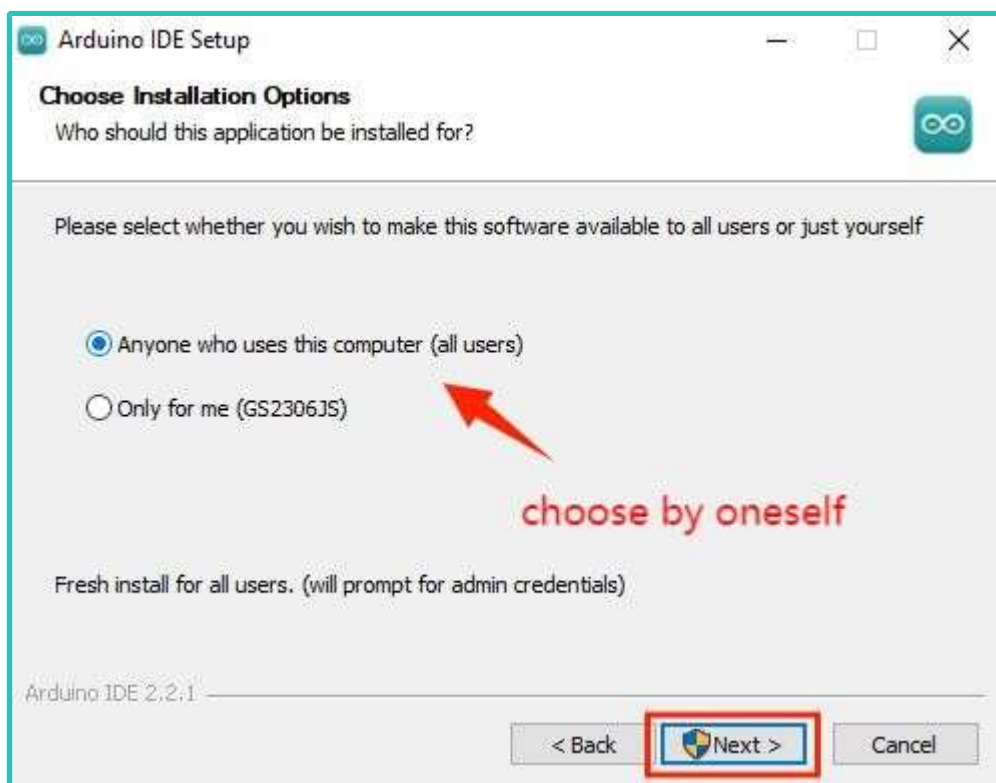


2. Po instalaci se zobrazí následující obrazovka a vyberte možnost "I Agree".

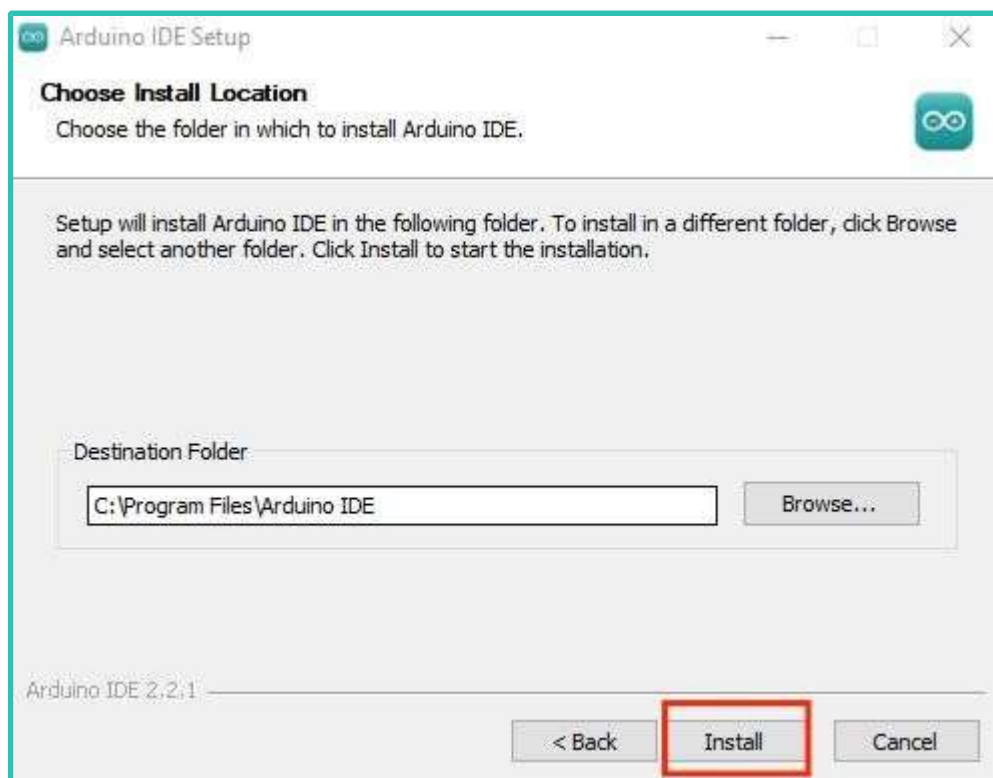


3. Po kliknutí na tlačítko "I Agree" se zobrazí následující obrazovka a klikněte na tlačítko "Next".

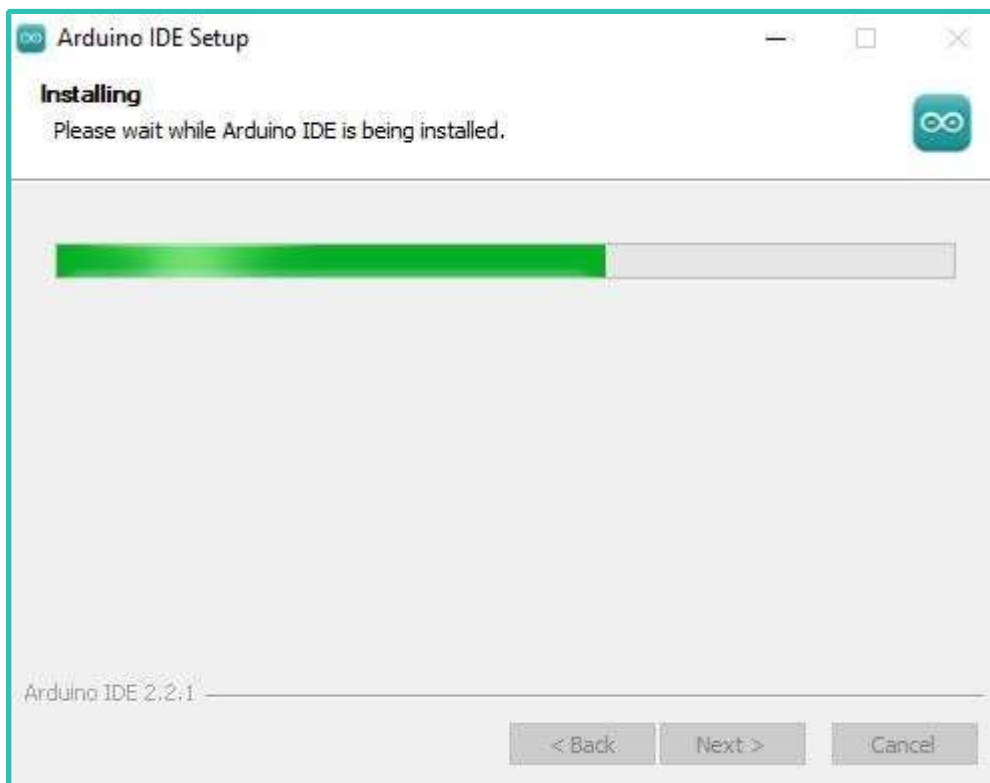




4. Po kliknutí na tlačítko "Další" se zobrazí následující obrazovka.  
Klikněte na tlačítko "Instalovat".



5. Instaluje se prostředí Arduino IDE.



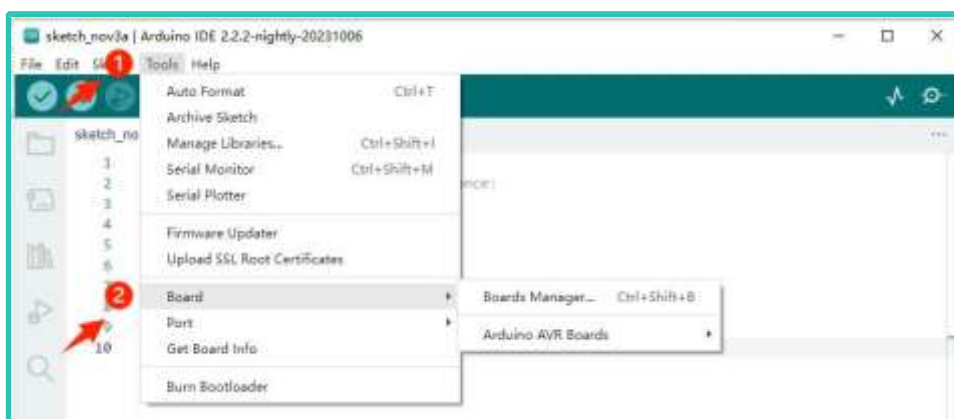
6. Instalace je dokončena.



7. Najděte na ploše zástupce prostředí Arduino IDE a dvojklikem na něj otevřete prostředí Arduino IDE.



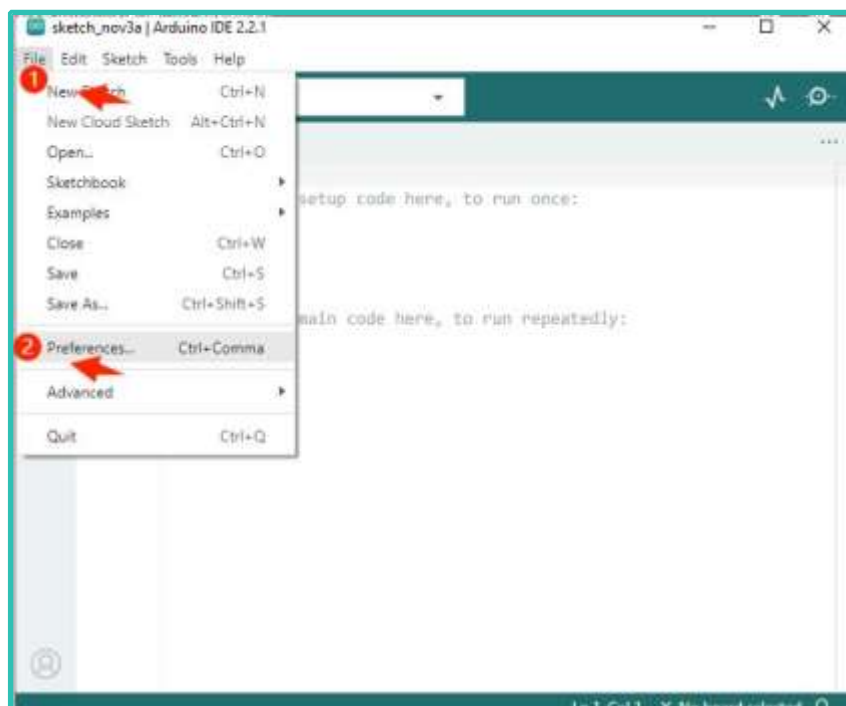
Krok 3: Přidání zdrojů řídicí desky pro ESP32 v prostředí Arduino IDE. Při programování v prostředí Arduino IDE je třeba zvolit vhodný model řídicí desky. Když otevřete Arduino IDE, vyberte "Tools" >"Board". Zjistíte, že Arduino IDE má pouze desky Arduino AVR a nemá esp32.



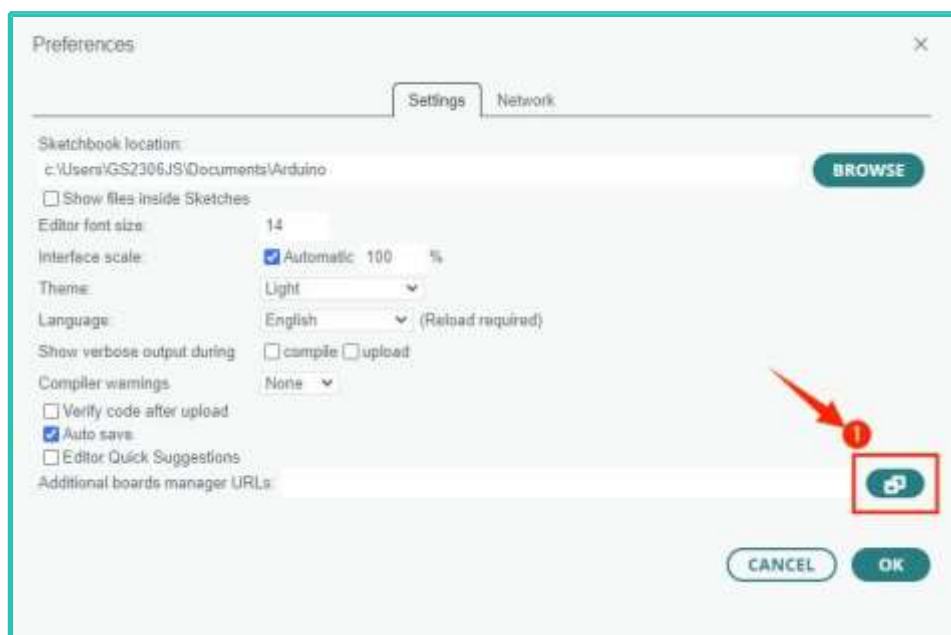


Dále podle níže uvedených kroků přidáme zdroje vývojové desky ESP32 do prostředí Arduino IDE.

1. Klikněte na "File"> "Preferences".



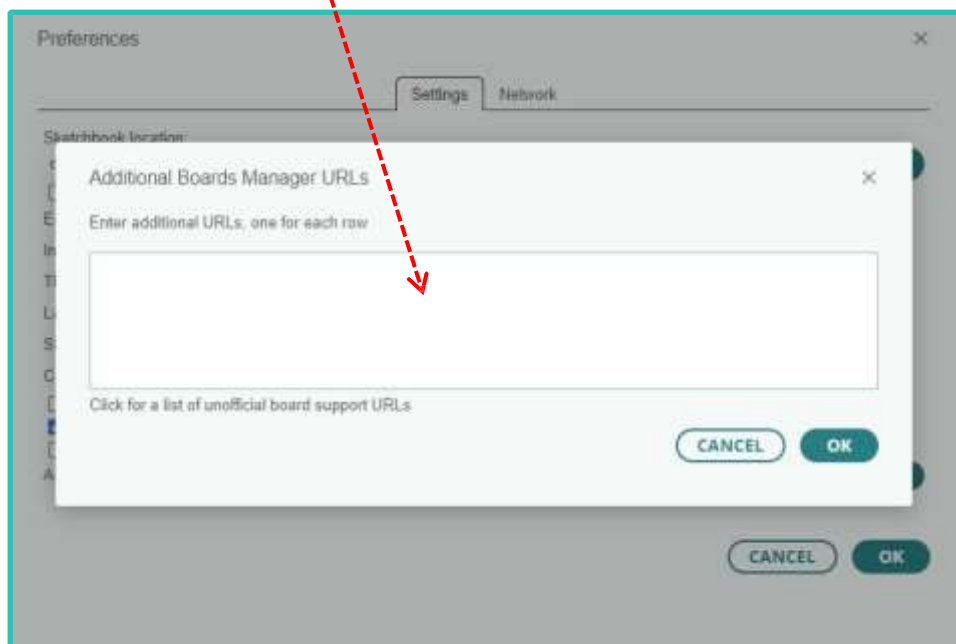
2. Přidání adres URL správce desek.





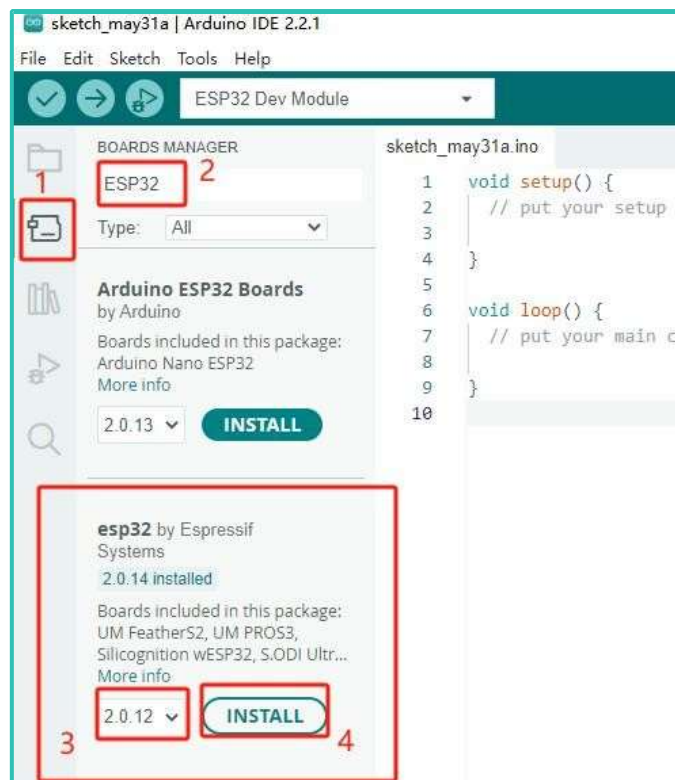
3. Přidejte odkaz v textovém poli níže do bílého pole označeného "Další adresy URL správce desek". Poté klikněte na tlačítko "OK".

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)  
[https://www.arduino.me/package\\_esp32\\_index.json](https://www.arduino.me/package_esp32_index.json)

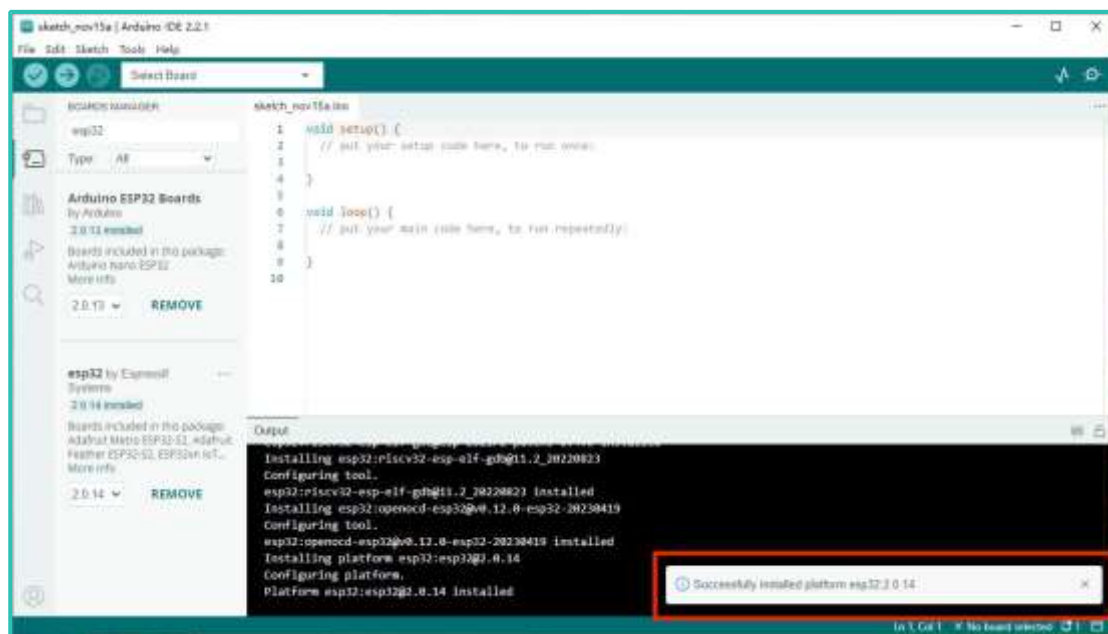


4. Kliknutím na ikonu desky na levém panelu ikon Arduina otevřete rozhraní "BOARDS MANAGER". Poté do vyhledávacího pole zadejte "ESP32". V zobrazených výsledcích vyhledávání vyberte verzi 2.0.12 zdrojů desky ESP32 a poté klikněte na tlačítko "INSTALL" pro instalaci.



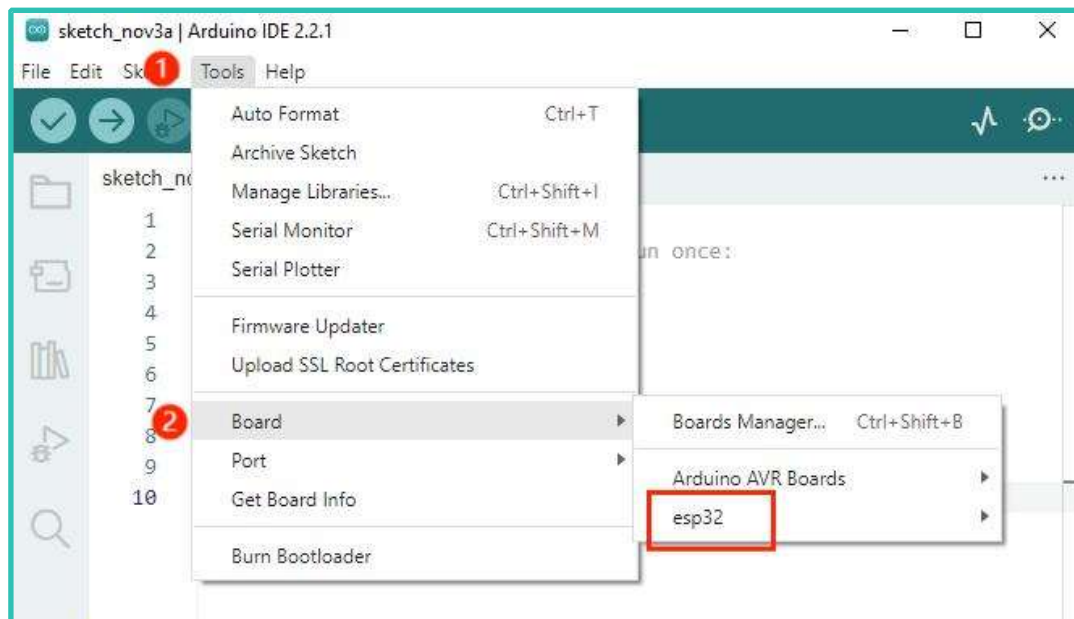


5. Po zobrazení následující obrazovky počkejte na dokončení instalace a zavřete Arduino IDE.



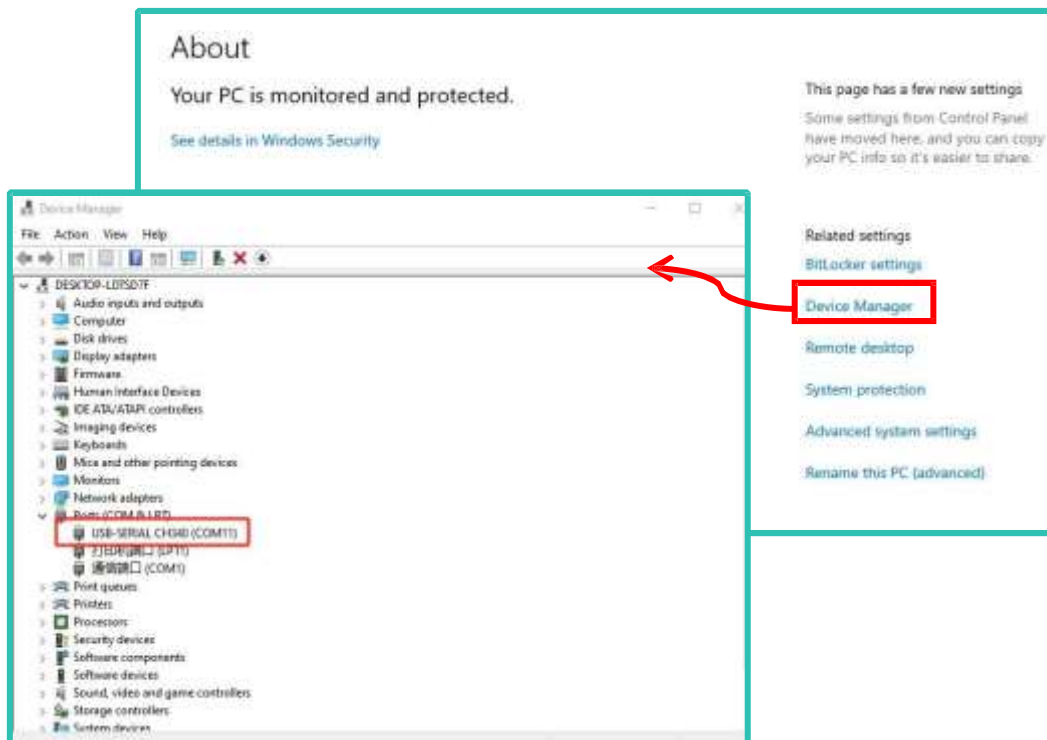
6. Znovu otevřete Arduino IDE a klikněte na "Nástroje" > "Deska" a zjistíte, že se objevila deska s řadičem esp32.



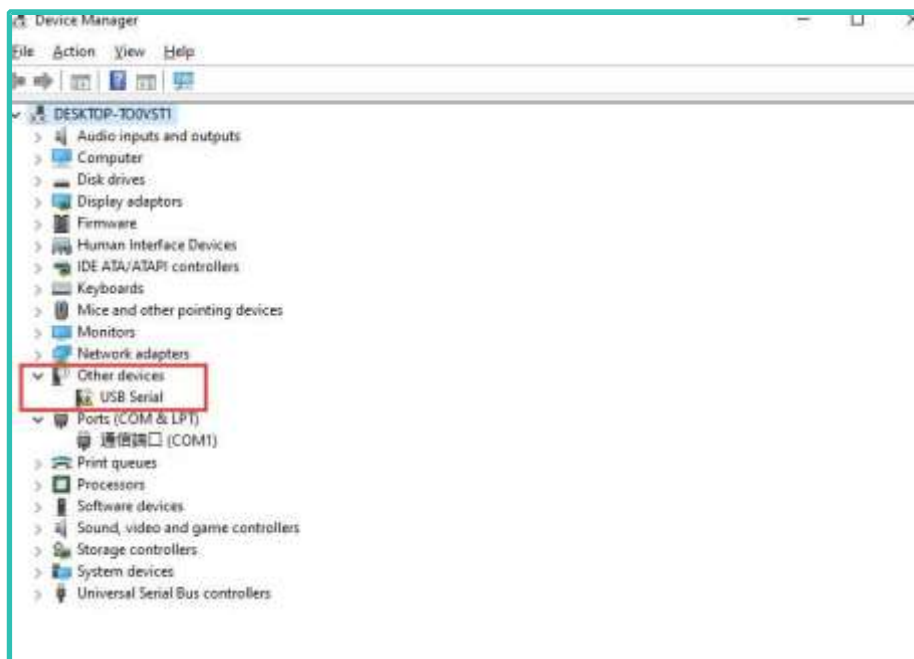


Krok 4: Otestujte vývojové prostředí. Po instalaci a konfiguraci vývojového prostředí Arduino IDE je třeba ověřit, zda může správně fungovat.

1. Připojte řadič ESP32 k počítači pomocí datového kabelu a poté v počítači vyhledejte Správce zařízení následující cestou: Správce zařízení: "Můj počítač" > "Atributy" > "Správce zařízení". Pokud ovladač normálně nainstalujete, najdete ve volbě "Port" položku "USB-SERIAL CH340 (COM11)" a obsah v závorce je sériové číslo portu. Toto číslo je náhodné a závisí na konkrétní situaci.



Pokud se ovladač nepodařilo nainstalovat, zobrazí se v možnosti "Další zařízení" položka "Neznámé zařízení".



### Tips

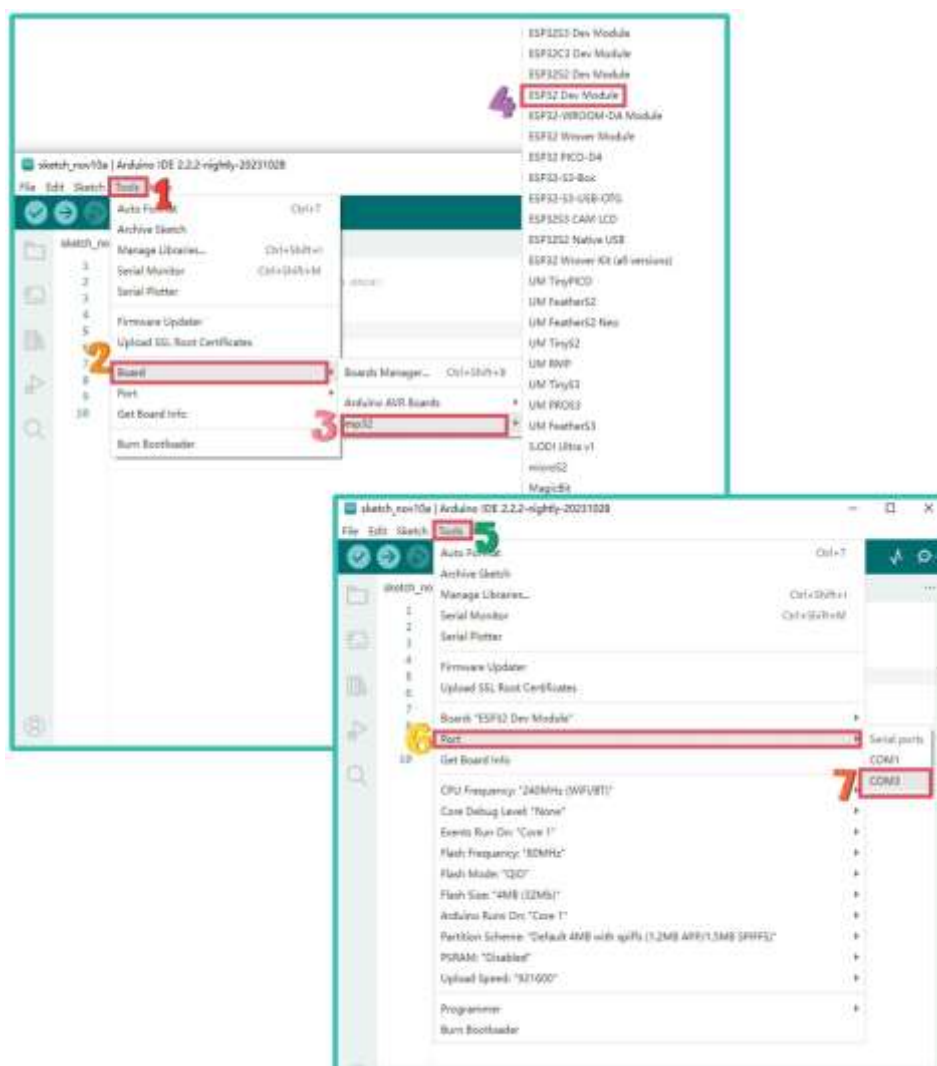
Pokud se vyskytne problém s řízením, na konci této učebnice najdete referenční řešení problému.





2. Nahrajte testovací program. Nahráním testovacího programu zkontrolujte, zda je proces nahrávání programu v pořádku. Otevřete Arduino IDE, vyberte odpovídající model řídicí desky ESP32 (ESP32 Dev Module) a odpovídající sériové číslo portu (stejně jako sériové číslo komunikačního portu ESP32 ve Správci zařízení).

Kliknutím na tlačítko "zkompilovat" zkontrolujte syntaktickou správnost

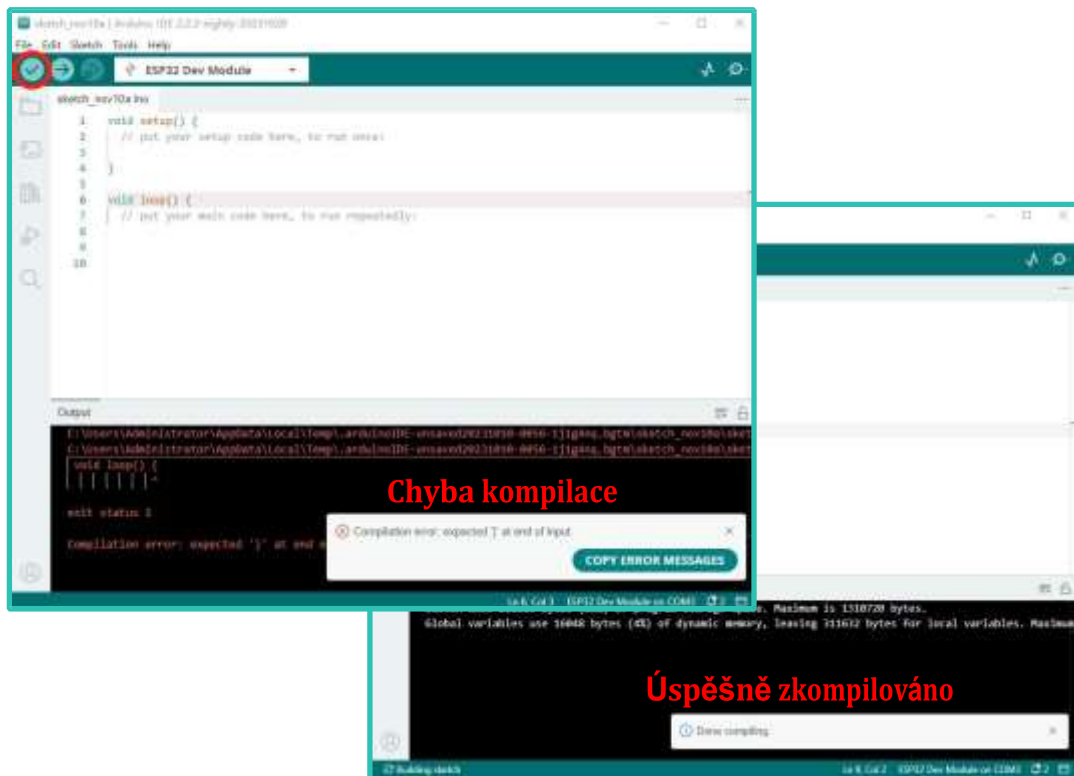


programu. Pokud je v programu syntaktická chyba, zobrazí se ve spodním okně chybové hlášení. Pokud se žádná chyba nevyskytuje

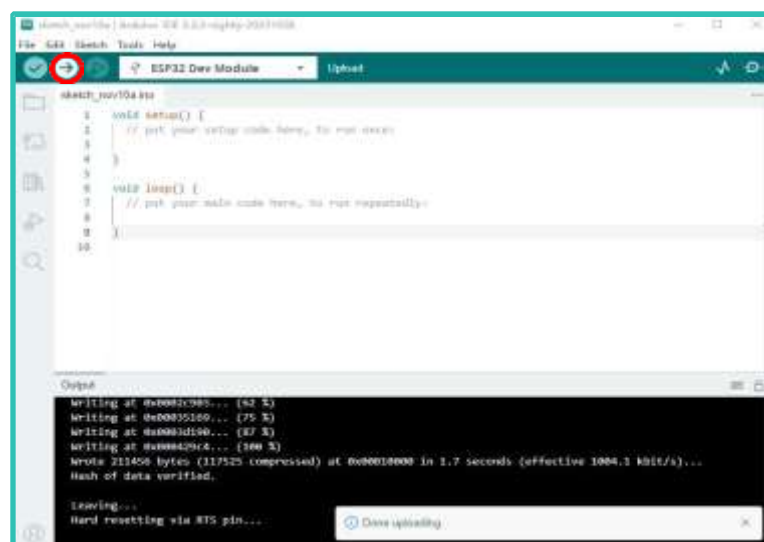




v syntaxi programu, kompilace je dokončena.



Po zkompileování můžete program nahrát kliknutím na tlačítko "Nahrát". Pokud se program úspěšně nahraje, zobrazí se zpráva "Done uploading".



**Již jsme se seznámili se základním vývojem prostředí řídicí desky ESP32. Dále ji využijeme k vytvoření vlastní chytré domácnosti.**





## Smart Home Lab

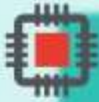


### Asistent chytré domácnosti



#### Popis projektu

Vytvořte pro svou chytrou domácnost domácího pomocníka a nechte ho, aby se představil nebo jednoduše pozdravil uživatele.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name           | Quantity |
|---|----------------|----------|
|  | ESP32 Board    | 1        |
|  | USB Data Cable | 1        |

##### 2. Schéma zapojení hardwaru

Připojte řídicí desku ESP32 k počítači pomocí datového kabelu USB.





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. void setup() {
2.   //put your setup code here, to run once:
3.   //Open the serial port and set the baud rate
4.   Serial.begin(9600);
5. }
6.
7. void loop() {
8.   //Put your main code here, to run repeatedly:
9.   //Print the content you want to display:
10.  Serial.println("Hello,I'm LUMI,welcome to Smart Home!");
11. }
```

### 3. Popis instrukcí

`setup()` : Při spuštění programu se nejprve provedou instrukce uvnitř `setup()`. Obvykle se používá pro inicializaci proměnných, portů I/O a některých funkcí. Funkce `setup()` se spustí při zapnutí nebo resetu vývojové desky a spustí se pouze jednou.

`loop()`: Funkce `loop()` se spustí po funkci `setup()` a instrukce uvnitř ní se mohou provádět nepřetržitě.

`//`: Symbol pro jednořádkový komentář, který se používá pro komentování instrukcí v programu a sděluje ostatním, jak program funguje. Cokoli za symbolem `'//'` nebude zkompilováno a provedeno.

`Serial`: Nástroj pro sériovou komunikaci mezi řídicí deskou a počítačem nebo jinými zařízeními. Obsahuje mnoho instrukcí souvisejících se sériovou komunikací.

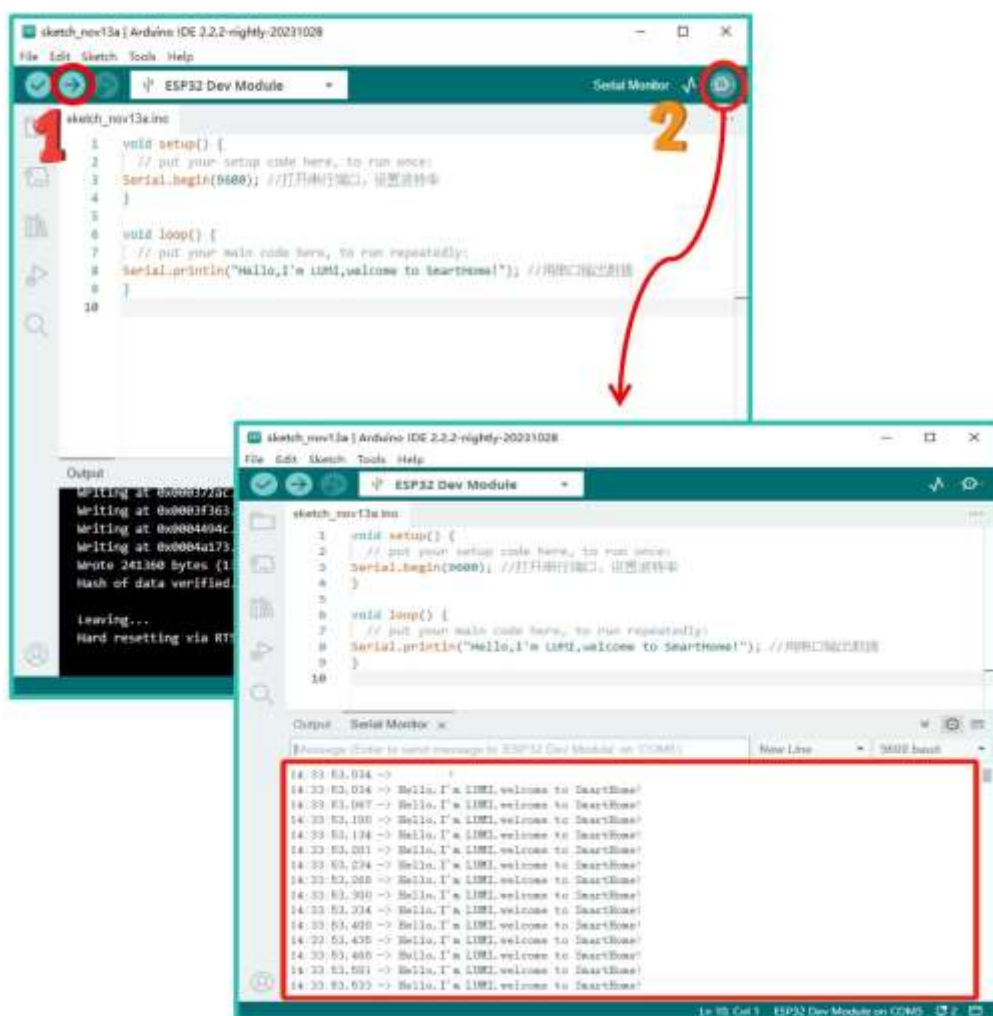
`Serial.begin(val)`: Otevře sériový komunikační port a nastaví rychlost přenosu dat (baud rate). `val` označuje hodnotu baud rate.

`Serial.println(val)`: Přenášená data mohou být libovolného typu. `val` označuje data, která mají být vytištěna.



## Tips

Když použijeme `serial.println( )` k vypsání vytištěných dat, můžeme si výstup z řídicí desky prohlédnout na sériovém monitoru prostředí Arduino IDE. Po nahrání programu klikněte na ikonu monitoru sériového portu, pod IDE se objeví okno monitoru sériového portu a zobrazí se data vypsaná sériovým portem.



## Kapitola 2 Rozsvi'te sv'uj domov

Když nastane noc, potřebujeme světla, která rozjasní tmu, abychom si mohli užívat noc v pohodlí a bezpečí našeho domova. V této kapitole se budeme zabývat tím, jak zlepšit pohodlí a úsporu energie v rodinném životě nastavením inteligentních systémů osvětlení.



## Oddíl 1 Rozsvícení světla

### Cíl učebního plánu

- ✎ Porozumět základním součástem obvodu
- ✎ Pochopení fungování LED diod
- ✎ Zvládnutí používání instrukcí digitálního výstupu portu ESP32 I/O
- ✎ Použití programovacích pokynů pro zprovoznění řídicí diody LED ESP32

**Jaké nástroje a metody používáte k osvětlení svého pokoje, když nastane noc?**

Od počátku lidské civilizace bojujeme proti tmě a snažíme se najít světlo a odstranit strach a nebezpečí, které tma přináší.

Po

nesčetné časy zkoumání a inovací, lidská osvětlovací technika se stále vyvíjí, zkušený z pochodní, olejové





na kurz elektrického osvětlení.

Elektrické světlo se zrodilo během druhé průmyslové revoluce jako člověkem vyrobené osvětlovací zařízení, které přeměňuje elektrickou energii na světelnou. Popularizace a použití elektrického světla výrazně zlepšily pohodlí lidského života. S pomocí elektrického osvětlení mohou lidé v noci vykonávat různé činnosti, jako je čtení, psaní a domácí práce. Kromě toho elektrické osvětlení také významně podporuje obchodní činnosti, průmyslovou výrobu a dopravu. Pod světlem elektrického osvětlení se výrazně podpořil hospodářský rozvoj a sociální pokrok lidí.

### Jak rozsvítit světlo?

Aby se elektrické světlo rozsvítilo, musí být zapojeno do obvodu. Základní obvod je uzavřená smyčka zdrojů energie, elektrických zařízení, vodičů, spínačů (nebo ovládacích zařízení obvodu).



**Napájení:** Zařízení, které dodává obvodu elektrickou energii. Mezi běžné zdroje napájení patří suché baterie, akumulátory, lithiové baterie, generátory atd. Obvykle měříme





energetický výkon zdroje pomocí jeho napětí, které se měří ve voltech (V).



**Elektrické spotřebiče:** Zařízení, která spotřebovávají elektrickou energii, ji přeměňují na jiné formy energie, jako jsou žárovky, motory, reproduktory atd.

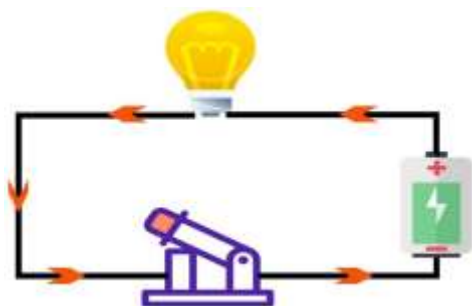


**Drát:** Vodič: spojovací vodič mezi zdrojem napájení a elektrickými spotřebiči, který je zodpovědný za přenos elektrické energie do elektrických spotřebičů. Běžnými materiály jsou měď, hliník atd.



**Přepínač:** Vypínač: Zařízení, které ovládá připojení a odpojení obvodu, a tím řídí provoz elektrického spotřebiče.

Pokud obvod tvoří uzavřenou smyčku, proud teče



v obvodu. Když proud projde lampou, získá lampa elektrickou energii a začne vyzařovat světlo.





## Terminologický výklad

### Elektrický proud

Ve vodiči se mohou volně pohybovat náboje. Po připojení zdroje se náboje ve vodiči působením napětí pohybují určitým směrem, čímž vzniká proud. Proud se dělí na střídavý a stejnosměrný, elektřina pro domácnost nebo průmysl je střídavá. Suché baterie, akumulátory, powerbanky apod. jsou stejnosměrné.



Stejnoseměrný proud teče z kladného konce zdroje, uzavřenou smyčkou a přitéká ze záporného konce zdroje. Jednotkou proudu je ampér (A).

S neustálým rozvojem vědy a techniky se snaha lidí o osvětlení již neomezuje na jednoduché funkce osvětlení, ale věnuje více pozornosti kráse, úspoře energie a inteligenci. Proto se také elektrické lampy neustále opakovaně vyvíjejí, od původních žárovek až po současné pokročilé LED lampy. V tomto kurzu se zaměříme na to, jak využít LED svítidla k návrhu inovativních osvětlovacích systémů.

### LED

LED dioda, známá jako světlo emitující dioda, je světelné zařízení vyrobené ze specifických polovodičových materiálů, které se široce používá v



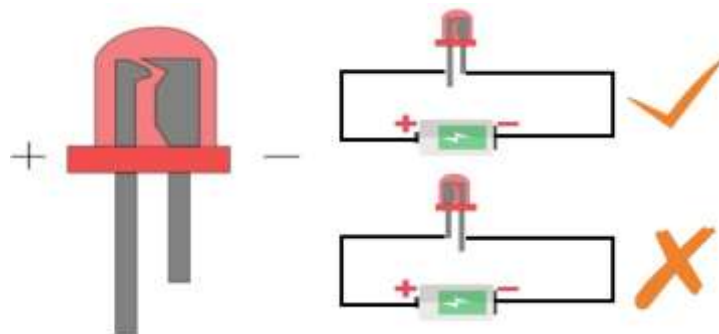
v oblasti osvětlení. Ve srovnání s žárovkami má LED zjevné výhody. Zaprvé je energeticky účinnější a šetrnější k životnímu prostředí, což může výrazně snížit spotřebu energie a snížit dopad na životní prostředí. Zadruhé, LED diody



reagují velmi rychle a dosahují maximálního jasu téměř okamžitě, zatímco žárovky potřebují určitou dobu k zahřátí. Kromě toho lze chromatičnost i jas LED světel nastavit podle individuálních potřeb, takže jsou flexibilnější a přizpůsobivější. Proto má použití LED v oblasti osvětlení širokou perspektivu.

Na rozdíl od jiných světel jsou světla LED s jedním vodičem, mají kladnou a zápornou elektrodu. Kladná elektroda LED musí být připojena ke kladnému konci zdroje napájení, záporná elektroda LED musí být připojena k zápornému konci zdroje napájení.





V tomto kurzu budeme používat integrační modul LED, který nám pomůže pohodlněji používat a chránit LED diody. Modul LED má tři vývody: Když je na pin S přivedena vysoká úroveň, dioda LED svítí, a když je na pin S přivedena nízká úroveň, dioda LED nesvítí.



**S:** Připojte k I/O pinu řídicí desky.

**V:** Připojte k 5V pinu řídicí desky.

**G:** Připojte ke kolíku GND řídicí desky.



### Tips

V řídicí desce ESP32 se úroveň HIGH vztahuje k napětí 5V nebo 3,3V a úroveň LOW se vztahuje k napětí 0V, které je v programu vyjádřeno jako high nebo low a může být také vyjádřeno jako 1 nebo 0.





## Smart Home Lab



### Rozsvícení LED



#### Popis projektu

Rozsviňte LED diodu, aby váš dům byl jasný, a navrhnete funkce osvětlení, jako jsou blikající LED diody.



#### Hardwarové schéma

1. Seznam hardwaru

| Picture   | name             | Quantity |
|---|------------------|----------|
|  | ESP32 Board      | 1        |
|  | White LED Module | 1        |
|  | USB Data Cable   | 1        |



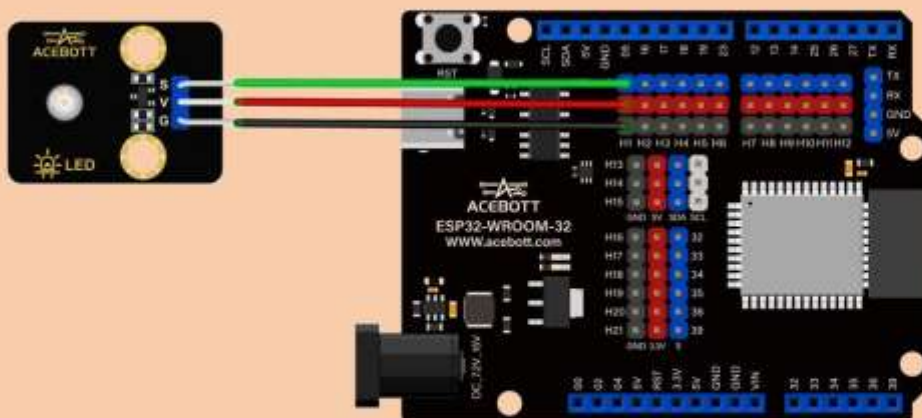


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 1 "01 Rozsvícení LED" montážního dokumentu.

## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. #define LED 5 //declare the number of the LED
2.
3. void setup() {
4.   // initialize pin LED as an output:
5.   pinMode(LED,OUTPUT);
6. }
7.
8. void loop() {
9.   //turn the LED on (HIGH is the voltage level);
10.  digitalWrite(LED,HIGH);
11.  delay(1000);//wait for a second
12.  //turn the LED off by making the voltage LOW:
13.  digitalWrite(LED,LOW);
14.  delay(1000);//wait for a second
15. }
```

### 3.1 Popis instrukcí

`#define`: Před kompilací programu se dané konstantní hodnotě přiřadí jméno, které lze později v programu použít místo konstantní hodnoty. Tato metoda může zlepšit efektivitu programování a čitelnost programů.

`pinMode(pin,mode)`: Nastaví zadaný pin I/O do vstupního nebo výstupního režimu, pin označuje zadaný pin I/O a jeho hodnota je zadané pořadové číslo pinu, mode označuje pracovní režim a jeho hodnota je INPUT nebo OUTPUT.

`digitalWrite(pin,value)`: pin představuje pořadové číslo pinu, value představuje hodnotu výstupního digitálního signálu a digitální signál má pouze dvě hodnoty: HIGH a LOW.

`delay(time)`: Zastavte program na nějakou dobu (time). 'time' je doba, po kterou se program pozastaví, v milisekundách.



#### Tips

Na řídicí desce ESP32 nemají všechny I/O piny funkci výstupu. Mezi nimi mají piny 34, 35, 36 a 39 pouze funkci vstupu, ale ne výstupu.

## Oddíl 2 Pulzující světla

### Cíl učebního plánu

- ✎ porozumět pojmu digitální a analogový signál
- ✎ Pochopení konceptu PWM
- ✎ Zvládnutí metody výstupního signálu PWM na ESP32
- ✎ Porozumět pojmu proměnné
- ✎ Zvládnutí používání proměnných
- ✎ Použití vln.PWM.k řízení jasu LED diod.



**Pokud se vám při večerním čtení doma zdá světlo příliš jasné, jak byste měli nastavit jeho jas?**

Když svítilnu používáme a vyměníme novou baterii, je světlo svítilny velmi jasné. Poté, co ji budeme nějakou dobu používat, bude světlo svítilny slabší a slabší. Tato je protože jak ji používáme, baterie je



klesá výkon a není schopen zajistit dostatečné napětí udržení jasu světla svítilny. Tento příklad nám jasně ukazuje



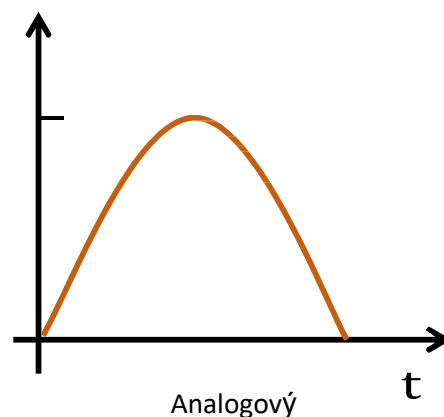
že jas světla souvisí s výkonem a napětím zdroje energie.

Pokud tedy měníme napětí LED, můžeme měnit jas LED. Jak změníte napětí?

V minulé lekci jsme ovládali zapnutí/vypnutí světla tak, že jsme pomocí příkazu `digitalWrite()` nastavili na výstupu I/O portu ESP32 buď 5 V, nebo 0 V. Tento typ signálu, který může ovládat pouze dvě hodnoty napětí, se nazývá digitální signál. Ve skutečnosti se hodnoty napětí mění kontinuálně a tento typ kontinuálně se měnícího signálu se nazývá analogový signál.

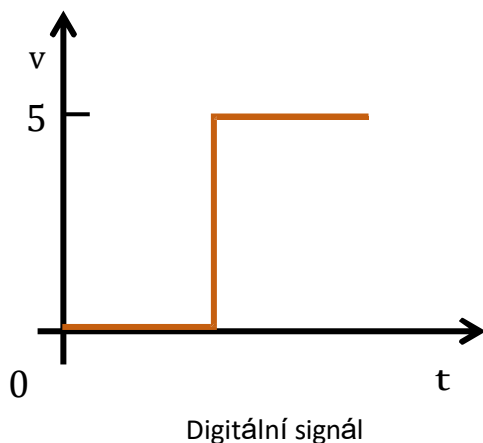
## Analogové a digitální signály

Analogový signál se týká na informace vyjádřený prostřednictvím a průběžně měnící se fyzikální veličinou, například například teplota, vlhkost, jas, proud, napětí atd. Často také nazýváme



analogový signál spojité signál. Může nabývat nekonečně mnoha různých hodnot v určitém časovém rozsahu. Například hodnoty analogového napěťového signálu na pravém obrázku mohou být 0 V, 0,1 V, ... 3,227 V, 4,666 V ... 5V.





Digitálním signálem se rozumí signál reprezentovaný diskrétními číselnými hodnotami, který se běžně používá k přenosu a ukládání informací.

na adrese elektronické zařízení nebo komunikační systémy. Hodnoty

digitálního signálu jsou konečné a odlišné v průběhu času. Například digitální signál napětí na levém obrázku má pouze dvě hodnoty: 5V a 0V.

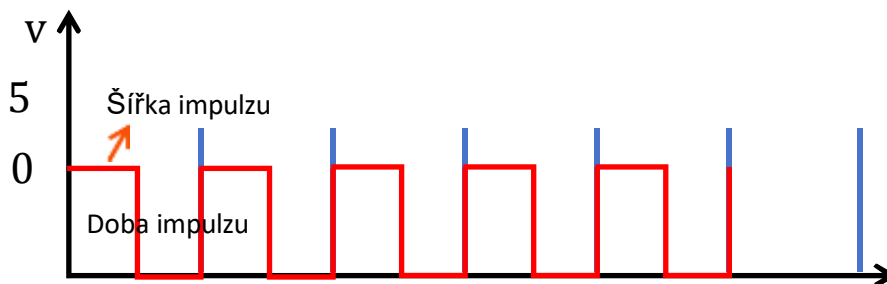
V ESP32 není možné přímo vyvést spojitý napěťový signál. Místo toho se k simulaci analogového napěťového signálu používá technologie PWM (Pulse Width Modulation).

### PWM

PWM, což je zkratka pro pulzně šířkovou modulaci, je technika, která používá digitální metodu k simulaci analogového signálu a k získání různých hodnot napětí.

Připojení (vysoká úroveň) a odpojení (nízká úroveň) obvodu lze přímo ovládat digitálními signály. Připojením a odpojením obvodu s vysokou frekvencí v rámci doby trvání lze získat pulzní signál, jak je znázorněno na následujícím obrázku.





Ve výše uvedeném grafu představuje červená čára pulzní signál. Vzdálenost mezi modrými čarami udává periodu pulzního signálu. V rámci každé periody se doba trvání vysoké úrovně nazývá šířka impulsu a poměr šířky impulsu k periodě se nazývá pracovní cyklus. Technologie pulzně šířkové modulace (PWM) funguje tak, že změnou šířky pulzu se mění konečné výstupní napětí.

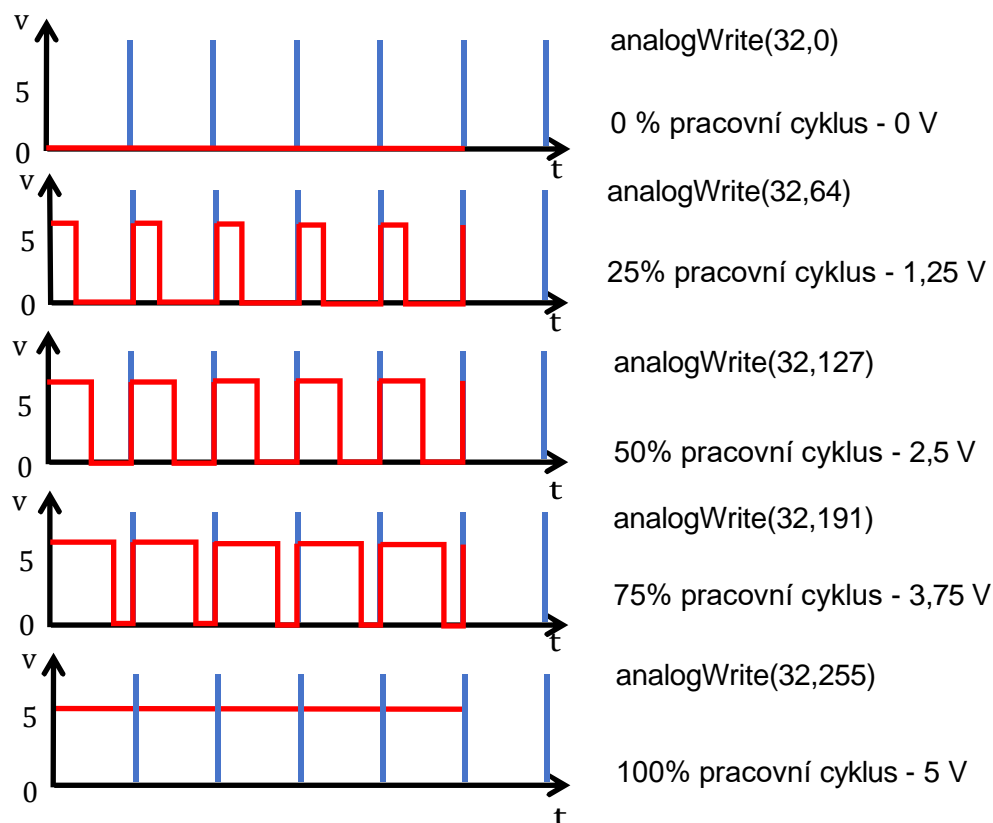
V prostředí Arduino IDE můžeme pomocí příkazu `analogWrite(pin, value)` vyvést signál PWM (Pulse Width Modulation) na zadaný pin. V tomto příkazu představuje parametr 'pin' číslo pinu, na který chceme vyvést signál PWM, a parametr 'value' představuje pracovní cyklus signálu PWM. Rozsah hodnot parametru 'value' je od 0 do 255, což odpovídá pracovnímu cyklu od 0 % do 100 %.

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





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



### Tips

Podle specifikací vývojové desky nejsou všechny I/O piny schopny vysílat signály PWM. Při použití výstupu PWM se musíme podívat do dokumentace vývojové desky, abychom zjistili, které piny mohou generovat signály PWM. V desce ESP32, kromě pinů 34 až 39, které nemohou generovat signály PWM, mají všechny ostatní piny schopné výstupu signálů funkci výstupu PWM.

Nyní jsme se naučili, jak vyvést více různých napětí. Jak ale můžeme v programu reprezentovat měnící se napětí? Pokud se v programu mění data, musíme použít proměnnou, která bude





uložit.

## Proměnná

Proměnná je v počítačovém programování kontejner, do kterého lze ukládat data a kdykoli je měnit.

Syntaxe proměnných v programování Arduino je :



**Datový typ** : Při vytváření proměnné nejprve deklaruje datový typ, který bude proměnná obsahovat. V programovacím prostředí Arduino patří mezi základní datové typy int, float, char a bool.

**Název** proměnné : Název proměnné představuje data uložená v proměnné, aby je bylo možné později v programu vyhledat a použít.

**Hodnota proměnné** : Jedná se o údaje uložené v proměnné. Program ukládá data do proměnných přiřazením symbolu '=' a může také použít symbol '=' k úpravě dat. Hodnoty proměnných lze kdykoli aktualizovat a upravovat podle potřeb programu.

Na předchozím obrázku program definuje proměnnou s názvem "lightness", která představuje jas světla, a přiřadí jí počáteční hodnotu 0.





## Smart Home Lab

### Pulzující světlo



#### Popis projektu

Změňte jas LED diody, nechte LED diodu pomalu přecházet z tmavé na světlou a poté z jasné na tmavou



#### Hardwarové schéma

1. Seznam hardwaru

| Picture   | name             | Quantity |
|---|------------------|----------|
|  | ESP32 Board      | 1        |
|  | White LED Module | 1        |
|  | USB Cable        | 1        |



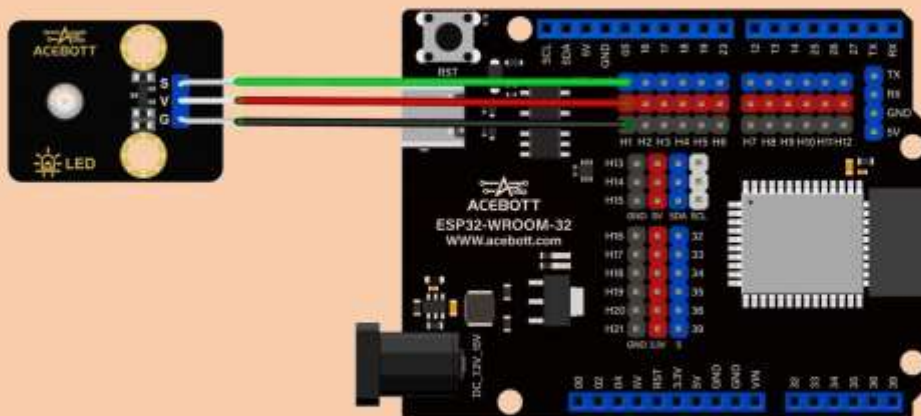


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 2 "02 Pulzující světlo" montážního dokumentu.

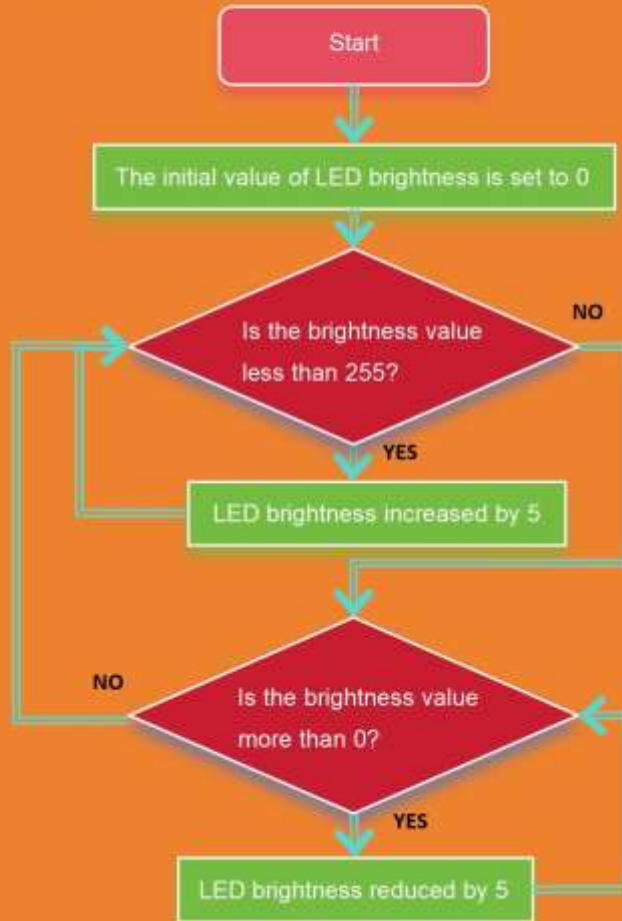
## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. #define LED 5 //declare the number of the LED
2. int lightness = 0; //initialize the brightness of the Light
3.
4. void setup() {
5.   pinMode(LED,OUTPUT); // initialize pin LED as an output.
6. }
```

```
7. void loop() {
8.   //gradually increase the brightness of the light
9.   for( lightness = 0;lightness<=255;lightness =lightness + 5){
10.    analogWrite(LED,lightness);
11.    delay(20);
12.  }
13.  //Gradually reduce the brightness of the light
14.  for( lightness = 255;lightness>=0;lightness =lightness -5){
15.    analogWrite(LED,lightness);
16.    delay(20);
17.  }
18. }
```

### 3. Popis pokynů

**for:** Příkaz for je instrukce smyčky, která opakovaně provádí určitou část instrukce. Syntaxe vypadá takto:  
for (Initial expression;loop conditions;Incremental expression){  
Příkazy;  
}

**Initial expression:** Slouží k inicializaci proměnné smyčky a spouští se pouze jednou před spuštěním smyčky.

**loop conditions :** Před každou iterací se zkontroluje podmínka smyčky. Pokud je podmínka pravdivá, provede se blok příkazů uvnitř kulatých závorek a inkrementační výraz. Pokud je podmínka nepravdivá, smyčka se ukončí.

**Statements (Příkazy):** Příkazy prováděné ve smyčce.

**Incremental expression:** Slouží k inkrementaci nebo dekrementaci hodnoty proměnné smyčky, vždy po provedení instrukcí bloku.



## Oddíl 3 Řízení světla (I)



### Cíl učebního plánu

- ✎ Pochopení fungování klíčů
- ✎ Ovládejte používání tlačítek
- ✎ Ovládání metody ovládání digitálních vstupů ESP32 I/O
- ✎ Zvládnout používání větrové struktury
- ✎ Tlačítkem můžete ovládat práci světla.



**Jak se ovládá osvětlení ve vaší domácnosti?  
Myslíte si, že existuje lepší a pohodlnější  
způsob?**



### Do It

Udělejte seznam způsobů, které jste viděli nebo byste chtěli zavést ovládání světel, a stručně popište postup zavedení.

| Už jste to viděli. | Čeho chcete dosáhnout? |
|--------------------|------------------------|
|                    |                        |
|                    |                        |
|                    |                        |





V předchozím kurzu jsme úspěšně implementovali spínání a nastavení jasu LED světel. Když však LED dioda pracuje, nemůžeme kontrolovat pracovní stav LED diody v reálném čase.



čas. K ovládní LED světel v reálném čase je třeba použít vstupní zařízení, jako jsou tlačítka nebo senzory. Tento typ vstupního zařízení může přenést náš záměr ovládat LED světla na vývojovou desku, čímž dosáhneme účelu ovládní v reálném čase.

## Tlačítko

Tlačítkem se rozumí mechanický spínač, který slouží k připojení nebo odpojení obvodu. Tlačítka se obvykle skládají z komponentů, jako je víčko tlačítka, pružina a kontakty.



Stisknutím víčka tlačítka se stlačí pružina, čímž se kontakty buď uzavřou, nebo rozepnou příslušný obvod. Tlačítka se hojně používají v různých elektronických zařízeních, jako jsou mobilní telefony, televizory, počítače a další.

V této lekci používáme modul tlačítkového obvodu, který se skládá ze zdroje napájení, tlačítka a odpovídajících elektronických součástek. Běžně se používá pro ovládní vstupů a výstupů v elektronických obvodech. Po stisknutí tlačítka se tlačítkový modul vysílá nízkou úroveň a po uvolnění tlačítka se tlačítko



modul vykazuje vysokou úroveň.



**S:** Připojte k I/O pinu řídicí desky. **V:** 5V/3,3V pin řídicí desky. **G:** Připojte kolík GND řídicí desky.



## Smart Home Lab

### Spínač světla



#### Popis projektu

Světla se zapínají a vypínají stisknutím tlačítka.



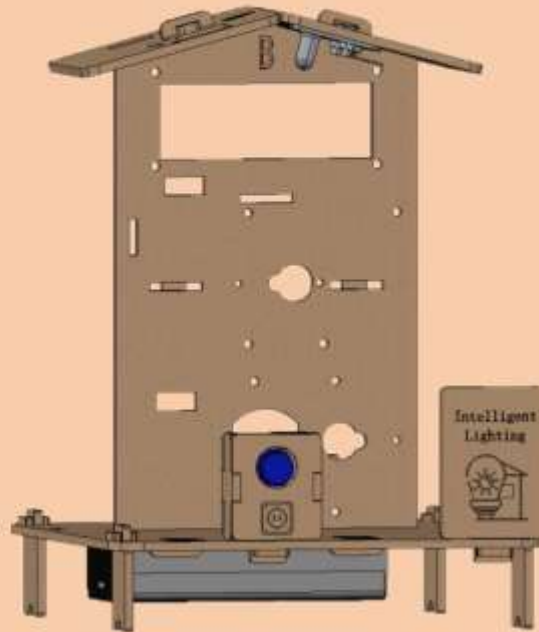
#### Hardwarové schéma

1. Seznam hardwaru

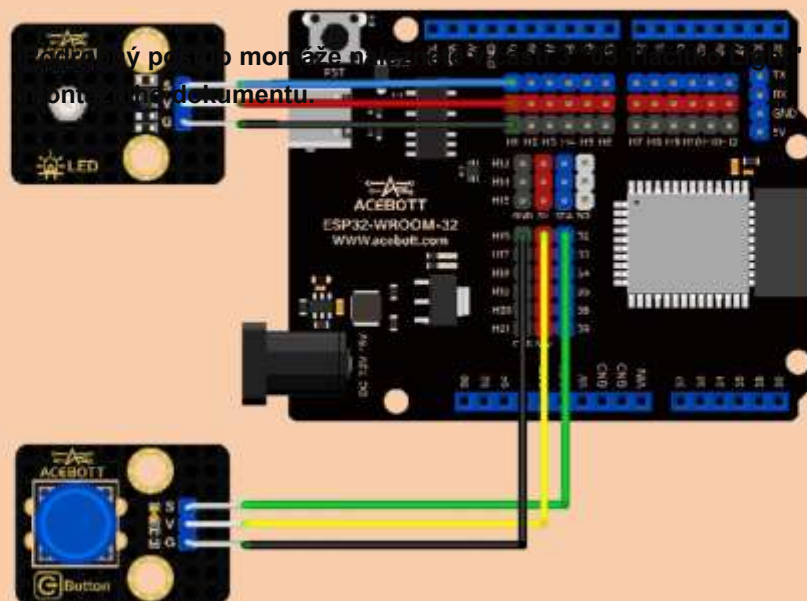
| Picture   | name             | Quantity |
|---|------------------|----------|
|  | ESP32 Board      | 1        |
|  | White LED Module | 1        |
|  | Button Module    | 1        |
|  | USB Data Cable   | 1        |



## 2. Hardware Structure Diagram



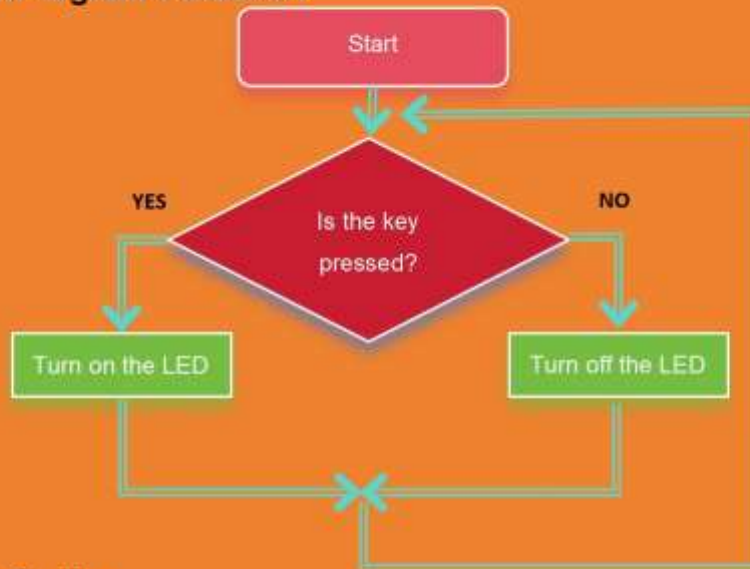
## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. #define LED 5 //declare the number of the LED
2. #define button 32 //declare the number of the button
3. int button_state = 0; //initialize the value of the button
4.
5. void setup() {
6.   pinMode(LED,OUTPUT); //initialize pin LED as output.
7.   pinMode(button,INPUT); // initialize pin button as input.
8. }
9. void loop() {
10.  //read the value of the button:
11.  button_state = digitalRead(button);
12.  if(button_state == 0){ //whether the button is pressed
13.    //the button is pressed:
14.    digitalWrite(LED,HIGH); //turn the LED on
15.  }
16.  else{
17.    //the button is not pressed:
18.    digitalWrite(LED,LOW); //turn the LED off
19.  }
20. }
```

### 3. Popis příkazu

**digitalRead(pin):** Po nastavení zadaného pinu do vstupního režimu lze tento příkaz použít ke čtení vstupní hodnoty zadaného pinu. Rozsah vstupních hodnot je 0 nebo 1. „pin“ označuje číslo portu zadaného pinu.

**if...else....** Příkaz if-else je podmíněná větvená struktura, která je detekována apredikcí. Syntaxe vypadá takto:

```
if(Podmíněný výraz){
```

```
  Příkaz 1;
```

```
  else{
```

```
    Příkaz 2;
```

```
  }
```

Pokud je podmíněný výraz pravdivý, provede se příkaz 1. Pokud je hodnota podmíněného výrazu nepravdivá, provede se příkaz 2.

**==:** Srovnávací operátory se používají pro srovnávací operace. '=' se používá k určení, zda jsou data na obou stranách operátoru stejná. Pokud jsou stejná, výsledek výrazu je pravdivý; pokud jsou



Nejsou-li stejné, výsledek je nepravdivý.

Mezi běžné srovnávací operátory patří:

**a!= b:** Zda a není rovno b

**a> b:** Zda a je větší než b

**a< b:** Zda a je menší než b

**a>= b:** Zda a je větší nebo rovno b

**a<= b:** Zda a je menší nebo rovno b



### **Optimalizace schématu**

Optimalizujte funkci kontrolní kontrolky klíče tak, aby po stisknutí a uvolnění klíče kontrolka svítila a po dalším stisknutí a uvolnění kontrolka zhasla.





vzdálenost, pohyb apod, a poté tyto informace převádí na elektrické signály nebo digitální data. Senzory hrají klíčovou roli v různých oborech a aplikacích s širokou škálou využití od vědeckého výzkumu přes průmyslovou výrobu, zdravotnictví až po každodenní život.

Mezi běžné senzory patří světelné senzory, senzory teploty a vlhkosti, barevné senzory, zvukové senzory, infračervené senzory, plynové senzory atd. Principy těchto senzorů se liší, ale všechny mají společný cíl: převádět fyzikální jevy na data, která mohou být využita elektronickými zařízeními nebo počítači. To umožňuje monitorovat, řídit, automatizovat a shromažďovat data. Vývoj sensorové techniky měl zásadní vliv na moderní technologie a průmysl a přinesl do našeho života mnoho vymožeností a inovací.

V této lekci se nejprve seznámíme s dotykovými senzory a vyzkoušíme si ovládání LED pomocí dotykových senzorů.

## **Dotykový senzor**

Dotykový snímač je zařízení používané ke sledování dotykových událostí a obvykle se skládá z kapacitní desky, která detekuje přítomnost objektu sledováním změny kapacity mezi objektem a kapacitní deskou. Kapacita se vztahuje ke schopnosti uchovávat





mezi dvěma vodiči a jeho velikost je ovlivněna vzdáleností a plochou mezi vodiči. Když se k desce kondenzátoru přiblíží nějaký předmět, změní rozložení elektrického pole na desce kondenzátoru, což způsobí velikosti kondenzátoru. Dotykové snímače využívají tento princip ke sledování událostí dotyku objektů.

Modul dotykového senzoru použitý v této lekci se skládá ze zdroje napájení, dotykového panelu a odpovídajících elektronických součástek. Když se dotykového panelu dotkne ruka, modul vydá nízkou úroveň; když se dotykového panelu nedotýká žádný předmět, modul vydá vysokou úroveň.



**S:** Připojte k I/O pinu řídicí desky.

**V:** Připojte 5V/3,3V pin řídicí desky.

**G:** Připojte kolík GND řídicí desky.





## Smart Home Lab

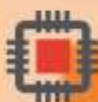


### Dotykové světlo



#### Popis projektu

Pomocí dotykového senzoru ovládáte fungování světla. Když se ruka dotkne dotykového senzoru, LED dioda se rozsvítí, když se ruka nedotkne dotykového senzoru, LED dioda zhasne.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name             | Quantity |
|---|------------------|----------|
|  | ESP32 Board      | 1        |
|  | White LED Module | 1        |
|  | Touch Sensor     | 1        |
|  | USB Data Cable   | 1        |

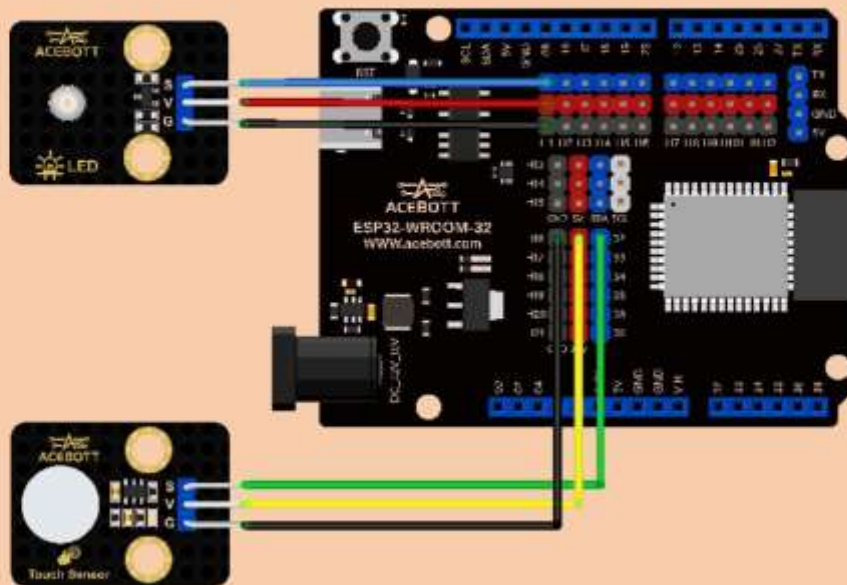


### 3. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 4 "04 Dotykové světlo" montážního dokumentu.

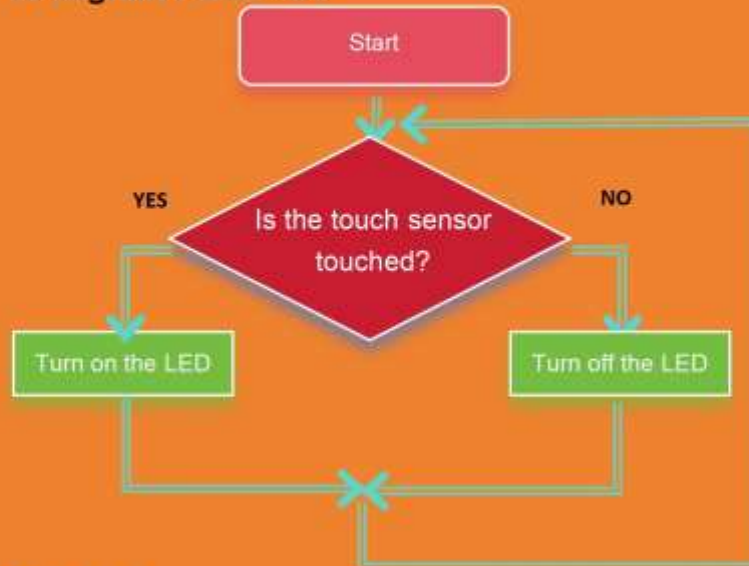
### 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. #define LED 5 //declare the number of the LED
2. #define touch 32 //declare the number of the touch sensor
3. int touch_state = 0; //initialize the value of the touch sensor
4.
5. void setup() {
6.   pinMode(LED, OUTPUT); //initialize pin LED as output.
7.   pinMode(touch, INPUT); //initialize pin touch sensor as input
8. }
9. void loop() {
10.  touch_state = digitalRead(touch);
11.  if(touch_state == 0){ //check if the touch sensor is touched
12.    digitalWrite(LED, HIGH); //turn the LED on
13.  }
14.  else{
15.    digitalWrite(LED, LOW); //turn the LED off
16.  }
```



### **Optimalizace schématu**

Pomocí dotykového senzoru se ovládá jas světla v rozsahu 0–255. Pokud se dotykový senzor nedotkne, jas světla je 0; při prvním dotyku dotykového senzoru je jas 80; při druhém dotyku je jas 160; při třetím dotyku je jas 255; při čtvrtém dotyku se jas vrátí na 0 a poté se opakují předchozí kroky.

## Oddíl 5 Chytré světlo

### Cíl učebního plánu

- ✎ pochopení principu světlo citlivých senzorů a lidských infračervených senzorů
- ✎ Ovládání analogového vstupu I/O portu ESP32
- ✎ Zvládnutí používání světlo citlivých senzorů a senzorů PIR
- ✎ Použití světlo citlivého senzoru a PIR senzoru k řízení práce LED diod



**Běžná světla musí být ovládána lidmi. Mohou světla fungovat samostatně bez lidského ovládání?**

Pokud máte systém chytrého osvětlení, který funguje sám o sobě, a jaké funkce byste chtěli, aby měl, napište své názory níže.



Světla by měla být během dne zhasnutá; když se setmí, měla by se rozsvítit. Aby se světla přes den automaticky vypínala a v noci automaticky zapínala, je to nutné .

Pomocí světelných senzorů noc.



### Světlo citlivý senzor



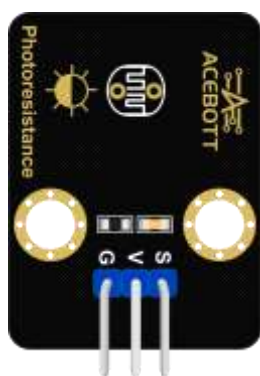
Fotorezistor, známý také jako rezistor závislý na světle (LDR), je senzor schopný detekovat intenzitu okolního světla. Jeho princip činnosti je založen na odporových vlastnostech





světlo citlivé materiály, obvykle vyrobené z polovodičových materiálů. Odpor fotorezistoru je nižší při silném světle a vyšší při slabém světle.

Fotorezistor, který v této lekci používáme, se skládá z rezistoru závislého na světle, zdroje napájení a některých dalších elektronických součástek. Snímač vysílá detekovanou intenzitu světla prostřednictvím vývodu S. Protože intenzita světla je analogový signál, je třeba tento pin připojit k analogovému I/O portu řídicí desky. Čím slabší je světlo, tím vyšší je hodnota přivedená na desku; čím je světlo jasnější, tím nižší je hodnota přivedená na desku.



**S:** Připojte analogový I/O pin řídicí desky.

**V:** Připojte 5V/3,3V pin řídicí desky.

**G:** Připojte pin GND řídicí desky.

#### Tips

V ESP32 jsou analogové vstupní piny pevně dané a omezené, z nichž 32, 33, 34, 35, 36, 39 atd. lze použít jako analogové vstupní piny a hodnota analogového vstupu je v rozsahu 0–4095.

Použití fotorezistorů v inteligentních osvětlovacích systémech umožňuje jejich automatické nastavení na základě změn denního světla.





a noční osvětlení. Tyto senzory však nejsou účinné při detekci přítomnosti osob. Pokud tedy v noci doma nikdo není, chytré osvětlení stále funguje, což vede k plýtvání energií. Abychom tento problém vyřešili, musíme zavést infračervený senzor člověka. Tento typ snímače dokáže účinně detekovat přítomnost osob, což umožní automatické nastavení inteligentního systému osvětlení na základě aktuálních potřeb a zabrání zbytečnému plýtvání energií.

### Infračervený paprsek

Infračervené záření (IR) je druh elektromagnetického záření, které se nachází pod spektrem viditelného světla a jehož vlnová délka se obvykle pohybuje mezi 0,7 mikrometru až 1000 mikrometrů.

Infračervené záření je generováno teplem

čím vyšší je teplota, tím silnější je vyzařované infračervené záření.

Infračervené záření patří do neviditelného světelného spektra a lidské oko ho nevidí. Lze jej využít pro aplikace, jako je komunikace na velké vzdálenosti, termální zobrazování, detekce objektů a měření teploty.



## Senzor PIR

PIR senzor je druh senzoru, který se používá speciálně k detekci lidské aktivity. Funguje tak, že když se lidé nebo zvířata pohybují, vyzařují infračervené záření. Lidský infračervený snímač dokáže citlivě snímat toto infračervené záření, aby určil, zda se do jeho snímacího dosahu dostal člověk nebo zvíře. Tento druh snímače má vysokou praktickou hodnotu a lze jej využít v mnoha oblastech, jako je monitorování bezpečnosti a systém Smart Home.

V této lekci PIR senzor, když je detekována lidská aktivita, pin S vyvede vysokou úroveň, a když není detekována žádná lidská přítomnost, pin S vyvede nízkou úroveň.



**S:** Připojte k analogovému I/O pinu řídicí desky.

**V:** Připojte k 5V/3,3V pinu řídicí desky.

**G:** Připojte ke kolíku GND řídicí desky.

Již jsme se seznámili s fotorezistorem a lidským infračerveným senzorem. Pojďme společně realizovat funkci chytrého světla!



## Smart Home Lab

### Inteligentní osvětlení



#### Popis projektu

Nechte LED diody pracovat autonomně podle okolního prostředí a realizujte automatické vypínání během dne a automatické zapínání v noci.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name                  | Quantity |
|---|-----------------------|----------|
|  | ESP32 Board           | 1        |
|  | White LED Module      | 1        |
|  | Photosensitive sensor | 1        |
|  | PIR sensor            | 1        |
|  | USB Data Cable        | 1        |



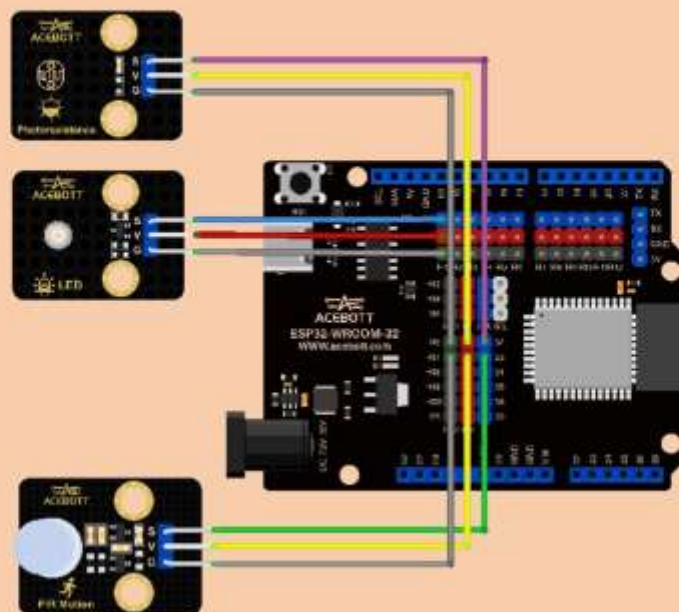


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 5 "03 Inteligentní světlo" montážního dokumentu.

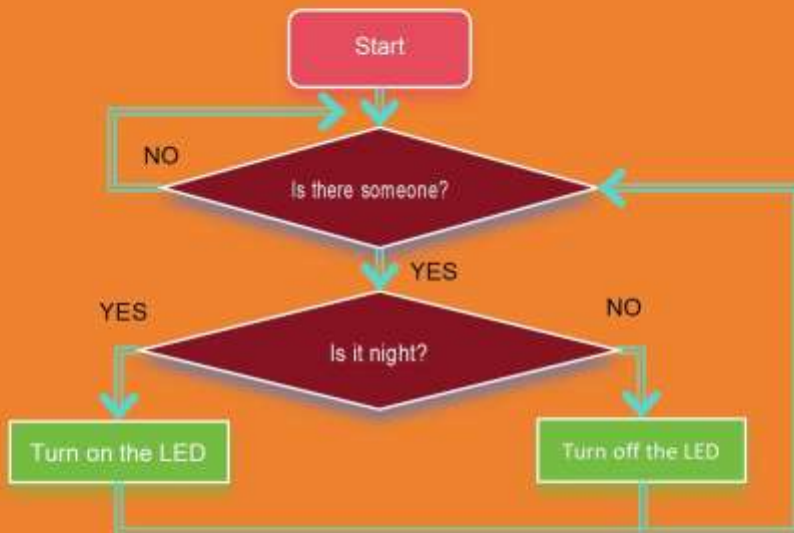
## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. coding

```
1. #define LED 5 //define the pins of LED.
2. #define PR 32 //define the pins of Photosensitive sensor
3. #define PIR 33 //define the pins of PIR sensor
4.
5. void setup() {
6.   pinMode(LED,OUTPUT);//set the LED to output mode
7.   pinMode(PR,INPUT);//set the photosensitive sensor to input mode
8.   pinMode(PIR,INPUT); //set the PIR sensor as the input mode
9. }
10.
11. void loop() {
12.   //get the Light intensity:
13.   int brightness = analogRead(PR);
14.   //get the human detection value:
15.   int check = digitalRead(PIR);
```

```
16. //Check if anyone is present:
17. if(check == 1){ //Someone is there
18.     //Set the value of the photosensitive sensor to exceed 3000
    as night, and check whether it is night:
19.     if(brightness>3000){//It's night
20.         digitalWrite(LED,HIGH);//Turn on LED
21.     }
22.     else{// It's daytime
23.         digitalWrite(LED,LOW);//Turn off LED
24.     }
25. }
26. else{//No one is here
27.     digitalWrite(LED,LOW);//Turn off LED
28. }
29. }
```

### 3. Popis instrukcí

**analogRead(pin):** Čte analogovou hodnotu ze zadaného analogového pinu. ESP32 má více měřicích kanálů, z nichž každý obsahuje 12bitový analogově-digitální převodník, který může mapovat hodnoty napětí 0-5 V na číselný vstup 0-4095.



#### Optimalizace schématu

Nechte jas světla měnit se podle intenzity okolního světla, čím silnější je okolní světlo, tím tmavší je LED světlo, a čím slabší je okolní světlo, tím silnější je LED světlo.

## Kapitola 3 Šťastný domov

Domáci zábava je klíčovým prvkem při budování harmonické a spokojené rodiny. Nejenže poskytuje členům rodiny příležitost k relaxaci a odpočinku, ale také podporuje rozvoj a upevňování blízkých vztahů. této kapitole se budeme zabývat tím, jak vybudovat inteligentní zábavní systém v inteligentní domácnosti, který vytvoří pohodlnější a příjemnější prostředí pro členy rodiny.





## Sekce 1: Duhový dům



### Cíl učebního plánu

- ✎ Prozkoumejte význam barev pro člověka.
- ✎ Porozumět základním barvám světla.
- ✎ Pochopení fungování světél RGB
- ✎ Zvládnout používání funkcí v programu
- ✎ Ovládněte používání světél RGB



**Proč je náš svět barevný? Co by se stalo, kdyby byl náš svět černobílý?**

Barvy hrají v lidském životě zásadní roli a v mnoha ohledech zásadně ovlivňují náš život. Použití různých barev může do našeho světa přinést nejen bohatý vizuální zážitek, ale také vyvolat různé emocionální



reakce a řídit chování lidí. Kromě toho jsou barvy často

slouží k předávání informací a log a jsou jim přiřazeny konkrétní





kulturní významy. Barva je víc než jen součástí vizuálního vnímání; má zásadní vliv na naše emoce, chování, kulturu a kvalitu života. Proto se barvy hojně využívají v různých oblastech, včetně umění, designu, medicíny, psychologie, reklamy a společenských věd.

### Proč je svět barevný?

Svět je barevný, protože náš systém zrakového vnímání vnímá různé vlnové délky světla v prostředí a tyto různé vlnové délky světla se vizuálně jeví jako různé barvy. Světlo, které je schopen vnímat lidský zrakový systém, se nazývá viditelné světlo a jeho vlnová délka se pohybuje mezi 400 nm a 700 nm.

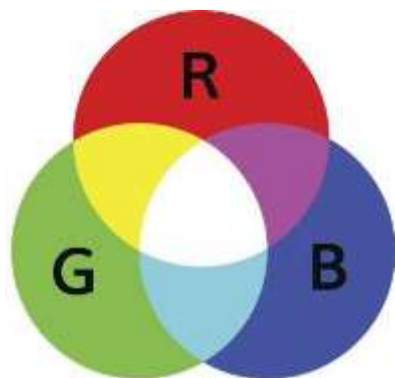
Barva předmětů, které vidíme, je určena materiálem povrchu předmětu.

Povrchy z různých materiálů absorbují některé vlnové délky světla, které vidíme. Například červené jablko absorbuje většinu barevného světla a odráží p



### Jak vytvořit barvy?





Ačkoli v přírodě existují různé barvy světla, které činí náš svět barevným, ve skutečnosti lze všechny tyto barvy vytvořit smícháním červeného, zeleného a modrého světla v různých poměrech. Světlo těchto tří barev - červené, zelené a modré - se obecně označuje jako tři základní barvy světla (RGB).

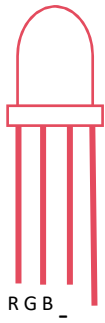
Smícháním světla těchto tří základních barev v různé míře lze získat světlo libovolné barvy. Tento způsob míchání se nazývá aditivní míchání barev. Například při smíchání červeného a zeleného světla ve stejné intenzitě vznikne žluté světlo. Pokud se přidá také modré světlo, může vzniknout mnoho dalších barev, například oranžová, fialová a azurová.

Princip tří základních barev RGB se široce používá v mnoha oblastech, zejména v televizi, počítačovém monitoru, barevném osvětlení a digitální fotografii.

V předchozích lekcích jsme k osvětlení místnosti použili světlo, které však vyzařovalo pouze světlo jedné barvy. Pokud chceme místnost během Vánoc barevnější, musíme použít světlo RGB, které vydává světlo více barev.



## Světlo RGB



Společná katoda

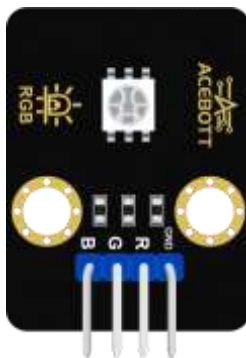


Společná anoda

Světlo RGB obsahuje tři malé LED diody, které vyzařují červené, zelené a modré světlo. Umožňuje míchat tyto barvy světla v různém jasu.

úrovni a proporcí pro vytvoření široké barev. Světlo RGB má čtyři vývody, přičemž nejdelší vývod je společná svorka a zbývající tři vývody ovládají činnost jednotlivých barevných LED.

V této lekci používáme modul RGB se společnou katodou, kde je společná svorka připojena k GND řídicí desky. Vývody R, G, B jsou připojeny každý k jinému I/O portu na desce řadiče. Když na každý z pinů R, G, B přivedena vysoká úroveň, rozsvítí se příslušná barva světla.



**G:** Připojte I/O port řídicí desky, zelená kontrolka svítí, když je vstup vysoký, jinak nesvítí.

**R:** Připojte I/O port řídicí desky, červená kontrolka svítí, když je vstup vysoký, jinak nesvítí.

**B:** Připojte I/O port řídicí desky, modré světlo svítí, když je vstup vysoký, jinak nesvítí.

**GND:** Připojte GND řídicí desky.



## Smart Home Lab



### Barevný dům



#### Popis projektu

Nastavte RGB světlo tak, aby cyklem procházelo různými barvami a vytvořilo efekt duhového světla.



#### Hardwarové schéma

1 .Seznam hardwaru

| Picture   | name           | Quantity |
|---|----------------|----------|
|  | ESP32 Board    | 1        |
|  | RGB Module     | 1        |
|  | USB Data Cable | 1        |



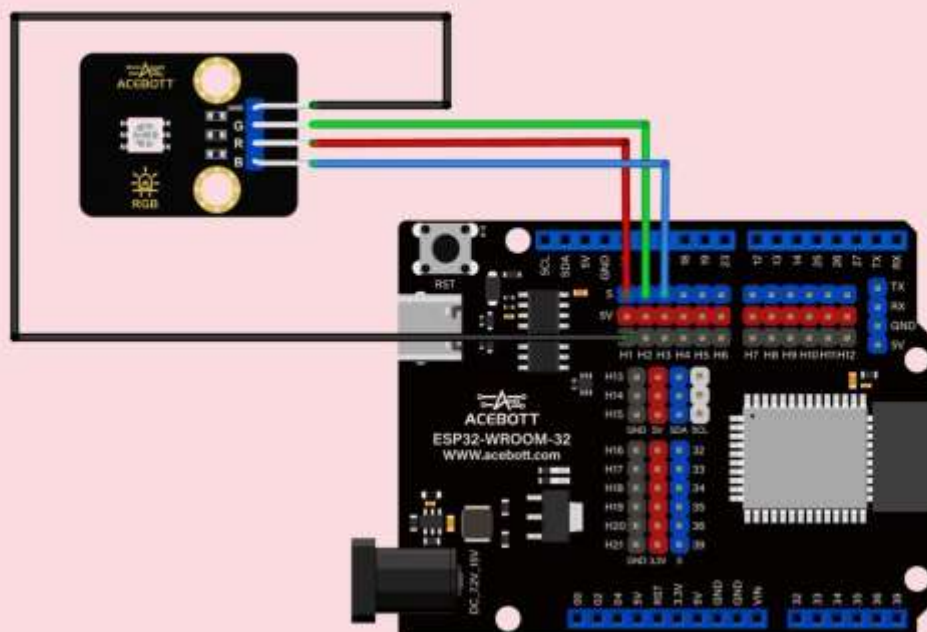


## 2. Hardware Wiring Diagram



Podrobný postup montáže naleznete v části 6 "06 Barevné světlo" montážního dokumentu.

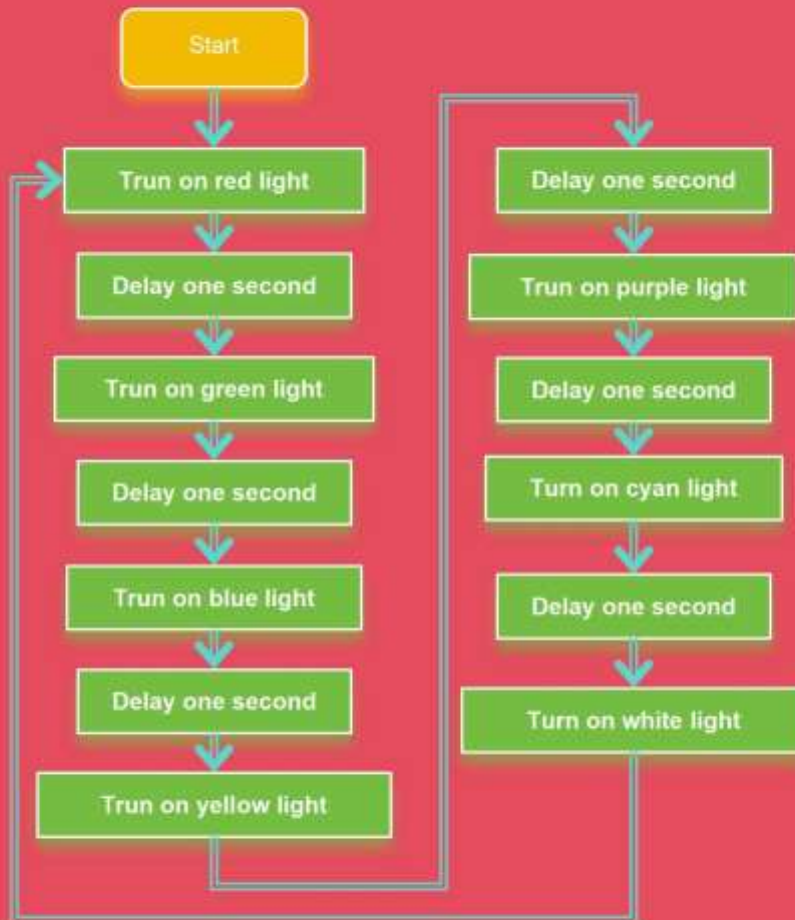
## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



## 2.Coding

```
1. #define R 5 //define the pins of red light of RGB
2. #define G 16 //define the pins of green Light of RGB
3. #define B 17 //define the pins of red Light of RGB
4.
5. void setup() {
6.   pinMode(R,OUTPUT); //set the pin of red light to output mode
7.   pinMode(G,OUTPUT); //set the pin of green Light to output mode
8.   pinMode(B,OUTPUT); //set the pin of blue light to output mode
9. }
10.
11. void loop() {
12.   RGBColor(1,0,0); // turn on red light.
13.   delay(1000); //delay 1 second.
14.   RGBColor(0,1,0); //turn on green light.
15.   delay(1000); //delay 1 second.
16.   RGBColor(0,0,1); //turn on blue light.
17.   delay(1000); //delay 1 second.
18.   RGBColor(1,1,0); //turn on yellow light.
19.   delay(1000); //delay 1 second.
20.   RGBColor(1,0,1); //turn on purple light.
21.   delay(1000); //delay 1 second.
22.   RGBColor(0,1,1); //turn on cyan light.
23.   delay(1000); //delay 1 second.
24.   RGBColor(1,1,1); //turn on white light.
25.   delay(1000); //delay 1 second.
26. }
27.
28. //define the RGB Lamp color control function
29. //r, g, and b are the parameters that control the operation of each
30. //lamp in RGB with values of 0 or 1
31. void RGBColor(int r,int g,int b){
32.   digitalWrite(R,r);
33.   digitalWrite(G,g);
34.   digitalWrite(B,g);
35. }
```

### 3. Popis instrukce

**Funkce:** Funkce je část kódu v programu, která může provést konkrétní úkol a může být znovu použita. Moduluje kód, zlepšuje čitelnost programu a efektivitu programování. Funkce se dělí na knihovní funkce a vlastní funkce. Knihovní funkce jsou součástí základní knihovny Arduino, například `pinMode()`, `delay()` a další. Vlastní funkce je funkce, kterou vytvoří uživatel během programování. Syntaxe vlastní funkce je následující:

```
return Type functionName(parameterList){
functionBody;
return value;
}
```

**return Type:** Typ funkce je definován návratovou hodnotou funkce. Pokud funkce vrací celé číslo, je návratový typ `int`, pokud funkce vrací číslo typu `float`, je návratový typ `float`, a pokud funkce nevrací žádnou hodnotu, je návratový typ `void`.

**functionName:** Název funkce je identifikátor funkce. Funkce se volá podle svého názvu.

**parameterList:** Seznam parametrů se používá k přijetí vstupních parametrů funkce. Funkce mohou mít více parametrů nebo žádný parametr.



**functionBody:** Tělo funkce je konkrétní blok kódu, který funkce používá k provedení určité akce.

**return:** vrací specifický datový typ volajícímu funkci, přičemž datový typ musí odpovídat typu return před názvem funkce.



### Optimalizace schématu

Změňte barvu každého pixelu, aby RGB světlo mělo více barev.





## Oddíl 2 Elektronické kostky



### Cíl učebního plánu

- ✎ Porozumět konceptu náhodných čísel
- ✎ Zvládnutí metody generování náhodných čísel v Arduinu
- ✎ Pochopení principu fungování 8sekční digitální trubice
- ✎ Zvládnout používání čtyřmístné digitální trubice
- ✎ Použití algoritmu náhodných čísel a čtyřmístné digitální trubice k sestavení elektronické hrací kostky



**Hrál jsi někdy kostky? Které číslo na kostce má podle tebe nejvyšší pravděpodobnost?**

Kostky jsou běžným herním nástrojem, který

hry, strategie hry, a hraní rolí hry, poskytová náhodnost a zábava pro různé hry a zábavné aktivity. A

běžnou kostkou je šestistěnná kostka, ale existují i jiné typy kostek, například kuželové, čtyřstěnné, bílé a jiné tvary. Kostky jsou oblíbené díky své neurčitosti a náhodnosti. když hodíte kostkou





kostky, každá tvář stejnou šanci se objevit a výsledek nemůžete ovlivnit. Při každém hodu kostkou je výsledkem náhodné číslo.

V chytré domácnosti můžeme vyrobit elektronickou hrací kostku, která nahradí tradiční kostky a zvýší pocit zážitku a zábavy ze hry. Abychom mohli vyrobit elektronickou kostku, musíme nejprve vědět, jak vygenerovat náhodné číslo.

### Náhodné číslo

Náhodná čísla jsou hodnoty nebo výsledky, jejichž výběr nemá žádný zřetelný vzorec nebo předvídatelnost. Obvykle jsou nepředvídatelné, a proto mají široké využití různých oblastech, včetně informatiky, statistiky, kryptografie, herního designu, simulací a dalších. Náhodná čísla jsou dvojího druhu: pravá náhodná čísla a pseudonáhodná čísla.

**Skutečná náhodná čísla** jsou generována fyzikálními jevy, jako je házení mincí, házení kostkou, otáčení kolem, šum vzduchu, elektronický šum apod. Proces generování pravých náhodných čísel obvykle závisí na hardwarových zařízeních, takže je relativně bezpečný a není snadné jej předvídat nebo odhadnout.

**Pseudonáhodná čísla** jsou souborem číselných posloupností.

generované určitým algoritmem. Za určitých podmínek





mohou vykazovat podobné vlastnosti jako skutečná náhodná čísla. Pro generování pseudonáhodných čísel je třeba zadat počáteční hodnotu nebo semeno, které pomocí řady matematických operací vygeneruje posloupnost náhodných čísel. Při použití stejné hodnoty seedu se vygeneruje stejná posloupnost náhodných čísel, což znamená, že výsledná náhodná čísla jsou předvídatelná. Při použití generátoru pseudonáhodných čísel je proto třeba dbát na to, aby byla hodnota seedu náhodná, a zajistit tak dostatečnou náhodnost generovaných výsledků.

V počítačovém programování jsou generátory pseudonáhodných čísel běžným nástrojem pro simulaci náhodných událostí a generování náhodných čísel. V Arduinu používáme ke generování náhodných čísel dvě instrukce `randomSeed()` a `random()`. Kde `randomSeed()` slouží k nastavení náhodného semínka a `random()` slouží ke generování náhodných čísel v zadaném rozsahu.



#### Tips

V ESP32, když analogový vstupní pin není připojen k žádnému zařízení, jeho vstup je vykreslen jako náhodná hodnota. Můžeme se rozhodnout použít simulované vstupní hodnoty tohoto nepřipojeného zařízení jako náhodné semínko. Například:  
`randomSeed(analogRead(33));` // K pinu 33 není připojeno žádné zařízení

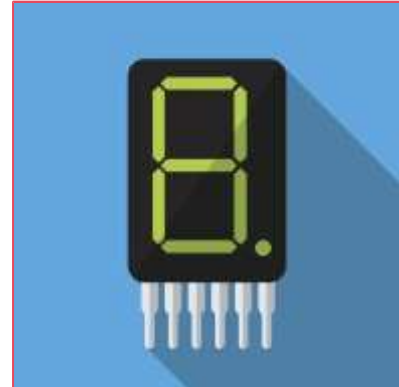
Nyní, když víme, jak vygenerovat náhodné číslo. Dalším krokem je zobrazení náhodného čísla, pokud je to vhodné. V této lekci použijeme k zobrazení náhodných čísel osmi segmentovou digitální trubici.





## 8 segmentů digitální trubice

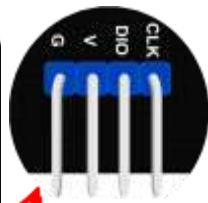
Digitální lampa je elektronické zobrazovací zařízení, které slouží k zobrazení znaků, jako jsou čísla, písmena a symboly, v digitální podobě. Běžně se používají v různých aplikacích pro počítání, měření, měření času a zobrazování. Osmi segmentová digitální trubice je jednou z nich



běžných typů digitálních trubic. Skládá se z osmi segmentů LED, z nichž každý představuje část digitálního displeje. Segmenty jsou obvykle uspořádány do tvaru "8" a každý segment může svítit nebo zhasínat a zobrazovat čísla 0 až 9 a některá písmena a symboly.

V tomto kurzu používáme čtyřmístný digitální trubicový modul, který se skládá ze čtyř osmi segmentových digitálních trubic a dokáže zobrazit více čísel a znaků. Modul používá k ovládní obsahu displeje komunikační protokol IIC, který může ušetřit prostředky pinů na vývojové desce.





**CLK:** Připojte port SCL (22) řídicí desky.

**DIO:** Připojte port SDA (21) řídicí desky.

**V:** Připojte 5V port řídicí desky.

**G:** Připojte port GND řídicí desky.



## Výklad terminologie

### Komunikace I2C

IIC (Inter-Integrated Circuit) je běžný komunikační protokol pro rozhraní mezi hardwarovými zařízeními, který lze také zapsat jako I2C. Cílem je mít co nejméně linek a co nejvyšší rychlost. Menší počet signálních linek může snížit rozměry pinů, což je důležité pro rané čipy.

Při použití rozhraní IIC je třeba připojit celkem čtyři vodiče, včetně: VCC, GND, SDA, SCL, z nichž SDA a SCL musí zabírat piny GPIO, které lze připojit k odpovídajícím pinům libovolné skupiny rozhraní IIC na vývojové desce. SCL je hodinová linka a řídí časování. SDA je datová linka, která přenáší data. Zjednodušeně řečeno, IIC potřebuje pouze 2 vodiče, může přenášet velké množství dat do více zařízení a snížit obsazení portů I/O na mikro kontroléru.



### Tips

Piny rozhraní IIC v ESP32 jsou 21 (SDA) a 22 (SCL).





Použití komunikace I2C k ovládní 4bitového digitálního elektronického modulu se nezdá být příliš složité?

Ano, jakékoli programování a manipulace, které zahrnují komunikaci, jsou složité. Abychom mohli pohodlněji ovládat modul 4bitové digitální trubice, můžeme k ovládní modulu použít příslušnou knihovnu Arduino.

### **Knihovna Arduino**

Knihovna Arduino je soubor předem napsaných programů, které umožňují implementovat konkrétní funkce voláním funkcí v knihovně. Prostředí Arduino IDE již obsahuje mnoho běžně používaných knihoven, například knihovnu Wire, knihovnu Servo atd. Tyto knihovny nám mohou pomoci rychle vyvíjet různé aplikace. Někdy však požadovaná funkce není ve vestavěných knihovnách k dispozici a v těchto případech je nutné nainstalovat vlastní uživatelské knihovny.

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

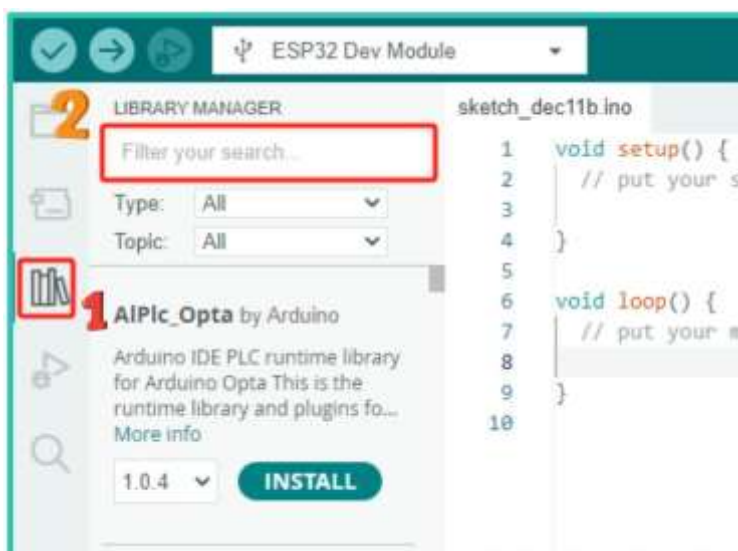
#### 1. Instalace online

Online instalace knihovny se provádí prostřednictvím prostředí Arduino IDE. Protože načítá zdroje knihoven z internetu, musí být váš počítač připojen k síti. Otevřete prostředí Arduino IDE, klikněte na ikonu zástupce správce knihoven vlevo a zobrazí se vyhledávací pole.

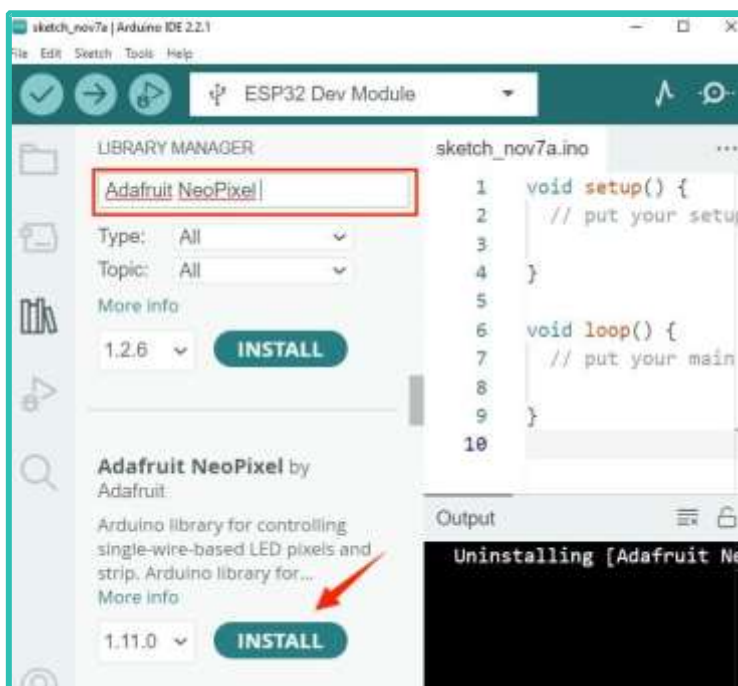




pro soubory knihovny.



Do vyhledávacího pole zadejte název knihovny, kterou chcete nainstalovat, a po stisknutí klávesy Enter se zobrazí příslušné výsledky hledání. V seznamu vyhledejte požadovanou verzi a klepněte na tlačítko instalovat. Po instalaci knihovny je třeba restartovat Arduino IDE, abyste ji mohli používat.

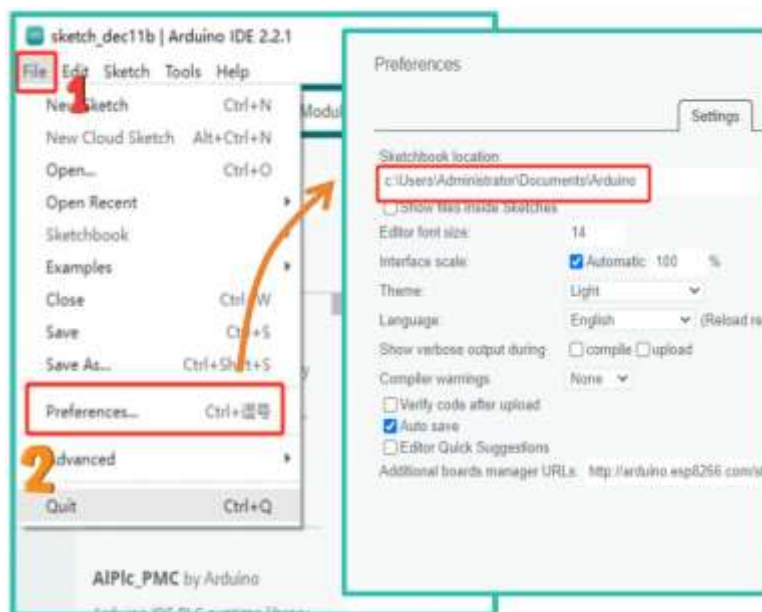


2. Místní instalace

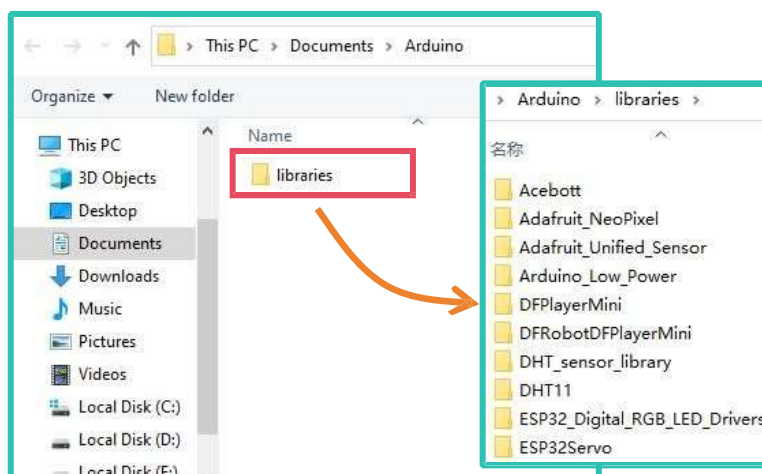




Místní instalace zahrnuje přidání zdrojů knihovny z místního úložiště počítače do vývojového prostředí Arduino IDE. Arduino IDE má výchozí umístění pro ukládání souborů a my můžeme jednoduše umístit zdroje knihovny do tohoto umístění. Výchozí adresu úložiště souborů prostředí Arduino můžeme zobrazit v části "Preferences" (Předvolby) prostředí Arduino IDE.



Otevřete cestu, kterou jsme si prohlédli výše, a pod cestou "Arduino" najdete složku "libraries". Do této složky 'libraries' můžete zkopírovat a vložit své knihovní soubory.





## Smart Home Lab

### Elektronické kostky



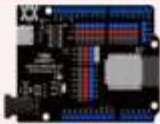



#### Popis projektu

Stisknutím tlačítka se na digitálním displeji zobrazí náhodné číslo od 1 do 6.



#### Hardwarové schéma

I. Seznam hardwaru

| Picture   | name                 | Quantity |
|---|----------------------|----------|
|  | ESP32 Board          | 1        |
|  | 4 digit digital tube | 1        |
|  | Button Module        | 1        |
|  | USB Data Cable       | 1        |



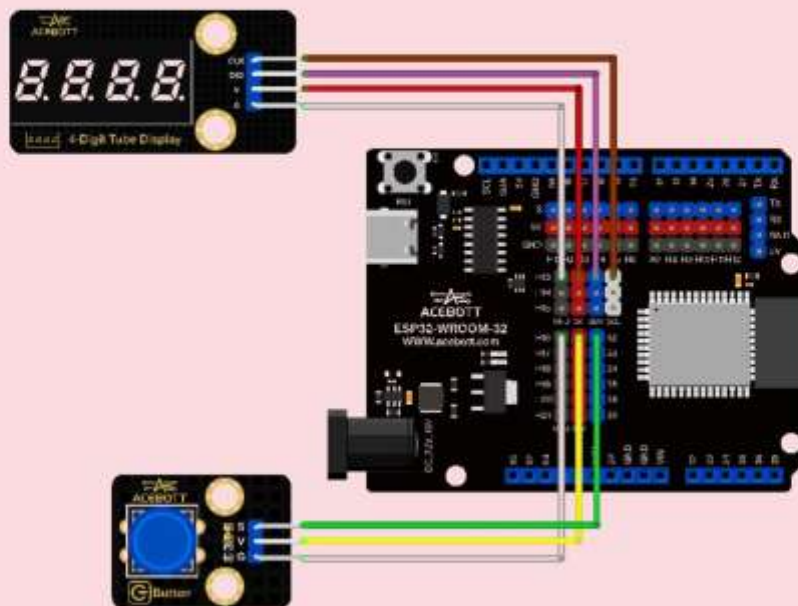


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 7 "07 Elektronické kostky" montážního dokumentu.

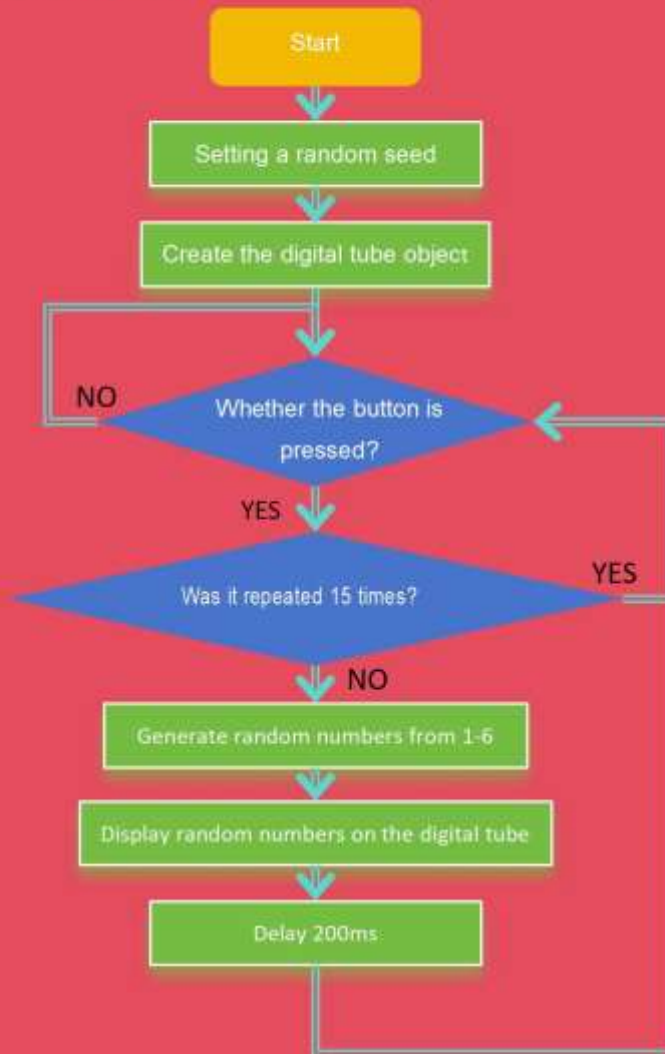
## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. coding

```
1. #include <Wire.h> //import the I2C Library
2. #include <TM1650.h> //import a Library of 4-bit digital tubes
3.
4. #define button 32 //define the pins for the button
```

```


5. int num = 0; //define a variable to hold a random number
6. char dis[4]; //create a character array to hold the string
7. TM1650 d; //create a TM1650 object
8.
9. void setup() {
10.   Wire.begin(); //enable I2C communication
11.   d.init(); //enable I2C communication
12.   d.setBrightness(TM1650_MIN_BRIGHT); //Set the display brightness
13.   d.displayOn(); //enable display
14.   int seed = analogRead(36); //get the random seed
15.   randomSeed(seed); //initialize the random generator
16.   pinMode(button, INPUT);
17. }
18.
19. void loop() {
20.   // put your main code here, to run repeatedly:
21.   if(!digitalRead(button)){ //check if the key is pressed
22.     delay(200); //delay detection to increase accuracy
23.     while(!digitalRead(button)); //wait for the button to release
24.     for(int a = 0; a < 30; a++){
25.       num = random(1,7); //generate random numbers 1-6
26.       itoa(num, dis, 10); //convert numbers to character arrays
27.       d.displayString(dis); //display random numbers on digit tube
28.       delay(100); //display time for each random number
29.     }
30.   }
31. }

```

### 3. Popis instrukce

**#include:** Slouží k zahrnutí knihoven mimo váš program, aby uživatelé mohli ve svých programech používat funkce a data knihovny. Knihovna I2C, která obsahuje řadu

**Wire.h:** funkcí a dat souvisejících s komunikací I2C. Každá knihovna Arduino se skládá ze souborů .h a .cpp. V souboru .h



Jsou deklarovány všechny funkce a typy proměnných a názvy v knihovně. Soubor .cpp obsahuje skutečný implementační kód pro všechny funkce. Chceme-li knihovnu zahrnout do našeho programu, stačí zahrnout hlavičkový soubor.

**TM1650.h:** knihovna 4místných digitálních trubíc, která obsahuje řadu funkcí pro ovládání 4místných digitálních trubíc.

**TM1650 d:** Vytvoří 4místný digitální trubkový objekt. TMI 650 je název třídy 4místného digitálního trubkového modulu. d je název objektu, který lze přizpůsobit. Data a funkce související s 4místným digitálním trubkovým modulem jsou zapouzdřeny v TM1650. Chcete-li tyto funkce použít, musíte vytvořit objekt a poté funkci z tohoto objektu vyvolat.


**d.init():** Funkce inicializace 4místného digitálního trubice.

**d.setBrightness():** Nastaví jas 4místného digitálního trubice

**d.displayOn():** Zapne funkci zobrazení 4místného digitálního trubice.

**d.displayString(array):** Zobrazí řetězec v zadaném znakovém poli na 4místném displeji. 'array' představuje znakové pole, které se má zobrazit.

**randomSeed(seed):** Inicializuje generátor náhodných čísel; 'seed' je náhodné semínko. Vygeneruje náhodné číslo mezi  $\text{random}(a,b)$ : a b, kde náhodné číslo zahrnuje dolní hranici a, ale nezahrnuje horní hranici b.



**itoa(num,array,scale):** Funkce pro převod celého čísla na řetězec. 'num' je celé číslo, které má být převedeno, 'array' je pole, do kterého se převedený řetězec uloží, a 'scale' je základ celé číslo po převodu.



### **Optimalizace schématu**

Při generování náhodných čísel nastavte náhodná čísla zobrazená na digitální trubici tak, aby se měnila z rychlých na pomalá a postupně se zpomalovala, až se zastaví.

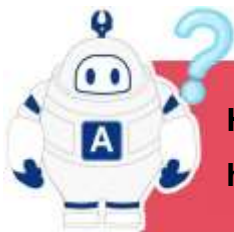


## ○ Sekce 3 Elektronický bzučák ○



### Cíl učebního plánu

- ✎ Prozkoumejte význam hudby pro člověka.
- ✎ Porozumět principům tvorby hudby.
- ✎ Pochopení principu bzučáku
- ✎ Zvládnutí používání bzučáku
- ✎ Přehrát hudební skladbu s bzučákem



**Hraješ na nějaký nástroj? Viš, jak vzniká hudba?**

Hudba je důležitou součástí našeho každodenního života. Je nejen formou zábavy, ale také důležitým nástrojem, který spojuje emoce lidí, šíří kulturu, podporuje sociální interakci a zlepšuje kognitivní schopnosti.



schopnost. Díky své mnohostranné roli hraje hudba v naší společnosti jedinečné a nezastupitelné postavení.



## Jak vzniká hudba?

Základem hudby je zvuk. Abychom pochopili původ hudby, musíme nejprve pochopit, jak zvuk vzniká. Zvuk vzniká chvěním nějakého předmětu. Když předmět vibruje, vzduch, který je s ním v těsném kontaktu, také vibruje, čímž vznikají zvukové vlny. Tyto zvukové vlny se šíří vzduchem a nakonec se dostanou do našich uší, což nám umožní zvuk vnímat.

Jako zdroje zvuku jsou definovány předměty, které mohou vydávat zvuk, například různé hudební nástroje. Různé zdroje zvuku vydávají zvuky o různých frekvencích a barvách.



prostřednictvím svých jedinečných způsobů. Některé nástroje vydávají zvuk vibracemi strun, dechové nástroje vydávají zvuk vibracemi vzduchu v trubicích a bicí nástroje vydávají zvuk údery nebo údery do povrchu.




Hudba se skládá z řady tónů a rytmů. Noty představují výšku tónů, přičemž různé noty odpovídají různým tónům.

zvukové frekvence. Rytmus řídí časování



vztahy mezi notami. Kombinace těchto tónů a rytmů tvoří melodii a rytmus, které jsou základními prvky hudby.

 **Tips**

|             |              |              |              |              |              |              |              |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| <b>Note</b> | <b>C3(1)</b> | <b>D3(2)</b> | <b>E3(3)</b> | <b>F3(4)</b> | <b>G3(5)</b> | <b>A3(6)</b> | <b>B3(7)</b> |
| <b>Freq</b> | <b>131</b>   | <b>147</b>   | <b>165</b>   | <b>175</b>   | <b>196</b>   | <b>221</b>   | <b>248</b>   |
| <b>Note</b> | <b>C4(1)</b> | <b>D4(2)</b> | <b>E4(3)</b> | <b>F4(4)</b> | <b>G4(5)</b> | <b>A4(6)</b> | <b>B4(7)</b> |
| <b>Freq</b> | <b>262</b>   | <b>294</b>   | <b>330</b>   | <b>350</b>   | <b>393</b>   | <b>441</b>   | <b>495</b>   |
| <b>Note</b> | <b>C5(1)</b> | <b>D5(2)</b> | <b>E5(3)</b> | <b>F5(4)</b> | <b>G5(5)</b> | <b>A5(6)</b> | <b>B5(7)</b> |
| <b>Freq</b> | <b>525</b>   | <b>589</b>   | <b>661</b>   | <b>700</b>   | <b>786</b>   | <b>882</b>   | <b>990</b>   |

Kromě tradičních hudebních nástrojů můžeme k přehrávání hudby využívat také elektronická zařízení. Dále se budeme zabývat tím, jak použít bzučák k přehrávání hudební skladby.

## **Bzučák**

Bzučák je běžná elektronická součástka používaná k produkci zvukových signálů. Princip činnosti bzučáku je založen na piezoelektrickém jevu nebo elektromagnetické indukci. Piezoelektrický





bzučáky obsahují piezoelektrický keramický prvek, který při přivedení napětí vibruje a vydává zvuk. Elektromagnetické bzučáky se skládají z elektromagnetické cívky a vibrující membrány, která rovněž vydává zvuk, když cívkou protéká proud.

Bzučáky se dělí na aktivní a pasivní. Aktivní bzučák může vydávat zvuk pouhým připojením ke zdroji napájení, zatímco pasivní bzučák potřebuje k vytvoření zvuku externí oscilační signál, obvykle řízený signálem čtvercové vlny s proměnnou frekvencí.

V této lekci se seznámíme s používáním pasivního bzučáku. Tento modul je charakteristický tím, že vydává zvuk odpovídající frekvenci čtvercového signálu přijatého na jeho signální vývod. Na platformě Arduino obvykle používáme funkci `tone()` pro generování signálu čtvercové vlny, který pohání pasivní bzučák a vydává zvuk.



**S:** Připojte I/O pin řídicí desky a vstupte signál čtvercové vlny.

**V:** Připojte 5V pin řídicí desky.

**G:** Připojte pin GND vývojové desky.





## Smart Home Lab

### Elektronický bzučák



#### Popis projektu

Napište program, který přehrává melodii „Twinkle twinkle little star“ přes bzučák.

#### Twinkle Twinkle Little Star

J = 100

1=C 4/4

||: 1 1 5 5 | 6 6 5 - | 4 4 3 3 | 2 2 1 - |

Twin-kle, twin-kle, li-ttle star, How I won-der what you are?



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name           | Quantity |
|---|----------------|----------|
|  | ESP32 Board    | 1        |
|  | P-Buzzer       | 1        |
|  | USB Data Cable | 1        |



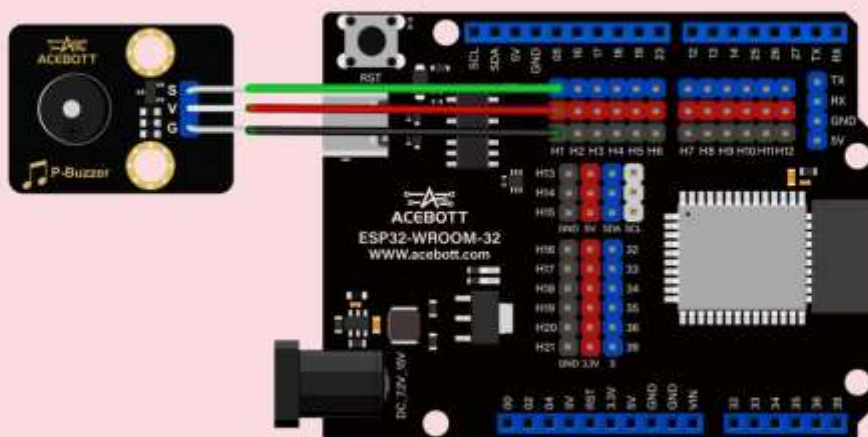


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 8 "08 Elektronický hudebník" montážního dokumentu.

## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. #define C4 262//define the frequency of each note:  
2. #define D4 294  
3. #define E4 330  
4. #define F4 350  
5. #define G4 393  
6. #define A4 441  
7. #define B4 495
```


```

8. #define buzzer 5 //declare the buzzer pin
9.
10. //store each note of Little Star:
11. int notes[] = {C4,C4,G4,G4,A4,A4,G4,
12.                F4,F4,E4,E4,D4,D4,C4};
13. //store the beat of each note of Little Star:
14. float beats[] = {1,1,1,1,1,1,2,
15.                  1,1,1,1,1,1,2};
16.
17. void setup() {
18.   pinMode(buzzer,OUTPUT); //set the buzzer pin to output mode
19.   Serial.begin(9600);
20. }
21.
22. void loop() {
23.   for(int i = 0;i < sizeof(notes);i++){//play each note in turn
24.     tone(buzzer,notes[i]);//play the note
25. //Each beat is 600ms, and each note is delayed by the
26. //corresponding time according to the corresponding beat number:
27.     delay(beats[i]*600);
28.     noTone(buzzer); //stop the note
29.   }
30.   delay(2000);//delay 2 seconds to Loop
31. }

```

### 3. Popis instrukce

tone(pin,frequency,duration): Ovládá specifikovaný pin tak, aby vydával obdélníkový signál specifikované frekvence a ovládá délku trvání obdélníkového signálu. 'pin' je číslo specifikovaného pinu, 'frequency' je frekvence obdélníkového signálu a 'duration' je délka trvání obdélníkového signálu. Pokud je parametr 'duration' vynechán, obdélníkový signál bude pokračovat donekonečna.



**noTone(pin):** Vypne výstup obdélníkového signálu na zadaném pinu.

**Array (pole):** Pole je datový typ používaný k ukládání více hodnot. Každý prvek pole má jedinečný index, který můžete použít k přístupu k prvkům pole. Syntaxe pro definování pole je následující:

```
type arrayName[arraySize];
```

„type“ je datový typ prvků v poli, „arrayName“ je název pole a „arraySize“ je velikost pole. Níže je uveden příklad:

```
int num[5];
```

Výše uvedený kód definuje pole obsahující 5 celých čísel, ke kterým lze přistupovat pomocí num[0] až num[4]. Pole můžete inicializovat při jeho definování. Například:

```
int num[5] = {1, 2, 3, 4, 5};
```

Výše uvedený kód definuje pole pěti celých čísel a inicializuje je hodnotami 1, 2, 3, 4 a 5. Pole můžete také inicializovat následujícím způsobem:

```
int = {1, 2, 3, 4, 5};
```

K prvkům pole můžete přistupovat pomocí indexů. Index začíná od 0, například:

```
int num[5] = {1, 2, 3, 4, 5};
```

```
int x = num[2];
```

Pole se často používají ve spojení s příkazy for, které pomocí smyčky procházejí všechny prvky pole. Například



```
int num[5] = {1, 2, 3, 4, 5};  
for (int i = 0; i < 5; i++) {  
    Serial.println(num[i])  
}
```

Výše uvedený kód přistupuje k všem prvkům pole postupně pomocí příkazu for.



### **Optimalizace schématu**

Umožněte přehrávání více skladeb a přepínání skladeb stisknutím tlačítek.



## ○ Oddíl 4 Chytrý hudební přehrávač ○



### Cíl učebního plánu

- ✎ Porozumět struktuře a principu modulu přehrávače MP3
- ✎ Pochopení principu sériové komunikace
- ✎ Ovládněte používání modulu přehrávače MP3
- ✎ Zvládnutí používání sériové komunikace ESP32
- ✎ Použití modulu přehrávače MP3 k vytvoření inteligentního systému přehrávače hudby



**Jaký hudební přehrávač obvykle používáte k poslechu hudby? Jaké jsou vlastnosti těchto přehrávačů?**

V každodenním životě je hudba bezpochyby nejlepším nástrojem k odbourání stresu. Ať už doma, v kanceláři nebo na cestách, hudba nám poskytuje bezpečné útočiště a pomáhá nám uvolnit se a relaxovat.



S rozvojem technologií se hudba přehrávače se staly v našem životě nezbytností. Nabízejí nejen různé hudební žánry, ale také podporují různé způsoby přehrávání hudby, čímž uspokojují naše rozmanité potřeby.

Existuje mnoho typů hudebních přehrávačů, včetně chytrých telefonů, tabletů, MP3, MP4 atd. Tyto přehrávače jsou vybaveny schopnostmi zpracování zvuku s vysokým rozlišením a mohou poskytovat



bezkonkurenční kvalitu hudby. Ať už se jedná o popovou, klasickou nebo lidovou hudbu, tyto přehrávače dokážou přesně reprodukovat původní zvuk hudby, díky čemuž máme pocit, že jsme přímo uprostřed ní.

Dále se zaměříme na představení přehrávače MP3 a prozkoumáme, jak použít modul pro přehrávání MP3 k návrhu inteligentního hudebního přehrávače.

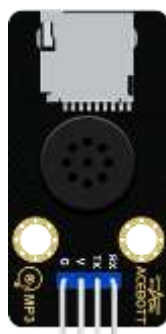
## Přehrávač MP3

Přehrávač MP3 je digitální zvukové zařízení určené speciálně k přehrávání hudby ve formátu MP3. Obvykle se skládá z mikroprocesoru, paměti, a



dekodéru zvuku . . . mikroprocesor slouží k řízení operací zařízení, paměť slouží k ukládání zvukových souborů a zvukový dekodér převádí zvukové soubory na slyšitelné zvukové signály. Kromě toho mají přehrávače MP3 další funkce, jako je přehrávání, pozastavení, zastavení, předchozí skladba, další skladba atd.

V této lekci používáme modul pro přehrávání MP3, který obsahuje nezávislý procesor, paměťovou kartu TF a reproduktor. Modul pro přehrávání MP3 můžeme ovládat a přehrávat hudbu prostřednictvím sériové komunikace v systému ESP32.



**RX:** RX: Připojí se k TX pinu řídicí desky a odesílá data do řídicí desky.

**TX:** TX: Připojí se k RX pinu řídicí desky a přijímá data z řídicí desky.

**V:** Připojte k 5V pinu řídicí desky.

**G:** Připojte ke kolíku GND řídicí desky.





## Tips

Před ovládním modulu pro přehrávání MP3 k přehrávání hudby je třeba nejprve uložit skladby na paměťovou kartu modulu MP3. Níže jsou uvedeny kroky pro uložení skladeb na modul pro přehrávání MP3:

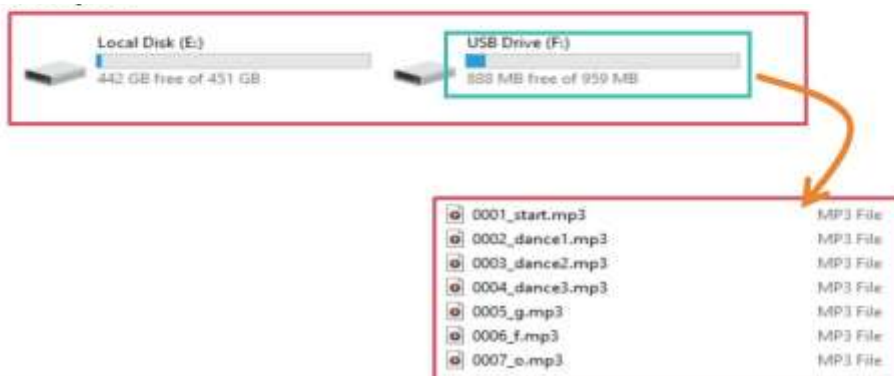
**(1) Vložte paměťovou kartu do slotu pro kartu modulu MP**



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

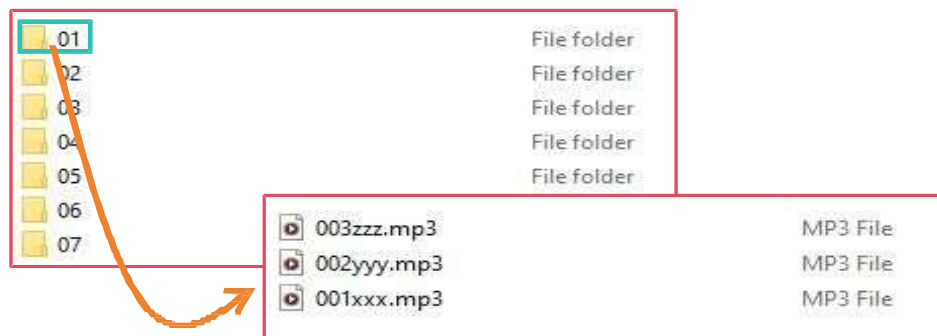


**(3) Po připojení modulu MP3 k počítači se na systémovém disku počítače zobrazí nová složka. Umístěte soubory MP3 do této nově zobrazené složky. Pokud jsou soubory MP3 umístěny v kořenovém adresáři paměťové karty, musí název souborů MP3 začínat čtyřmi číslicemi, například „0001 Hero.mp3“, „0002Jingle bells.mp3“.**



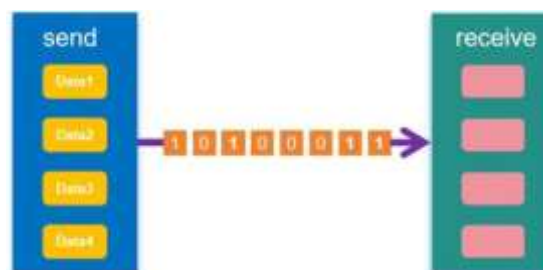


**(4) Pokud chcete vytvořit složku v kořenovém adresáři karty, založte složku s dvoumístným názvem a soubory MP3 ve složce. složku s třímístným názvem.**



## Sériová komunikace

Sériová komunikace (UART) je typ sériového protokolu pro přenos dat, který se používá pro výměnu dat mezi dvěma zařízeními. V sériové komunikace jsou data přenášena po jednom bitu v sekvenci s předem stanovenou přenosovou rychlostí.



Sériová komunikace se běžně používá pro výměnu dat mezi vestavěnými systémy, mikro kontroléry, senzory, akčními členy a dalšími zařízeními. Například komunikace mezi a .





mezi vývojovou deskou a počítačem nebo mezi senzorem a vestavěným zařízením lze dosáhnout prostřednictvím sériových portů. Tento způsob komunikace je jednoduchý a stabilní, takže je vhodný pro mnoho scénářů použití.

Sériová komunikace vysílá a přijímá data prostřednictvím dvou portů: TX a RX. TX je port pro vysílání dat, zatímco RX je port pro příjem dat. V ESP32 jsou k dispozici tři kanály sériové komunikace: UART0, UART1 a UART2, přičemž každému z nich jsou přiřazeny výchozí I/O piny.

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



### Tips

V ESP32 se pro stahování programů a ladění používá výchozí UART0. Pokud se pro komunikaci s externími zařízeními používá UART0, musí být externí zařízení připojené k UART0 před nahráním nebo laděním programu odpojeno. Piny pro tři sériové komunikační kanály na desce ESP32 nejsou pevně dané; lze je přizpůsobit tak, aby byly namapovány na libovolný pin.





## Smart Home Lab

### Inteligentní hudební přehrávač



#### Popis projektu

Vytvořte chytrý hudební přehrávač, který umožňuje pozastavit přehrávání, přehrávat, stříhat skladby a další všelijaké funkce.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name           | Quantity |
|---|----------------|----------|
|  | ESP32 Board    | 1        |
|  | MP3 Module     | 1        |
|  | Button Module  | 2        |
|  | Touch Sensor   | 1        |
|  | USB Data Cable | 1        |



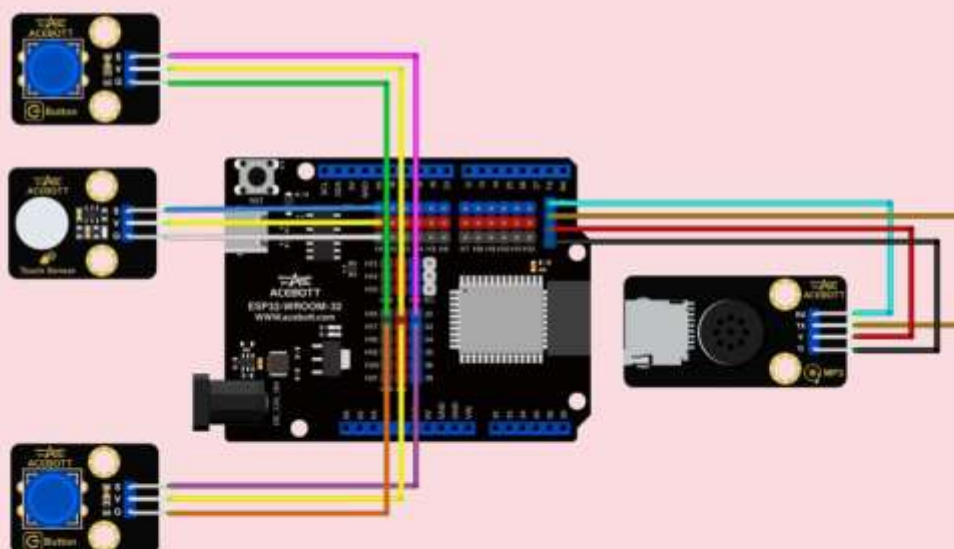


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 9 "09 Inteligentní hudební přehrávač" montážního dokumentu.

## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



## 2.Coding

```
1. #include <MY1690.h> //import the MP3 player Library
2.
3. #define touch 5 //declare the touch sensor's pin
4. #define button1 32 //declare the pin for key 1
5. #define button2 33 //declare the pin for key 2
6.
7. my1690 MP3; //generate an MP3 object
8.
9. void setup() {
10.   Serial.begin(9600); //set baud rate to 9600
11.   delay(100);
12.   MP3.setLoopMode(); //set loop mode
13.   MP3.play(); //play the song
14.   pinMode(touch, INPUT); //set the touch sensor pin to input mode
15.   pinMode(button1, INPUT); //set button 1 pin to input mode
16.   pinMode(button2, INPUT); //set button 2 pin to input mode
17. }
18.
19. void loop() {
20.   //detects if the touch sensor has been touched
21.   if(!digitalRead(touch)){//touch sensor is touched
22.     delay(200);
23.     while(!digitalRead(touch)); // Wait for the hand to leave
24.     //Play and pause commands. If the pause command is executed
25.     //now, the play command will be executed next time:
26.     MP3.playPause();
27.   }
28.   //check if key 1 is pressed
29.   if(!digitalRead(button1)){// button 1 is pressed
30.     delay(200);
31.     while(!digitalRead(button1)); //wait for button release
32.     MP3.previousTrack(); //play the last song
33.   }
34.   //check if key 2 is pressed
35.   if(!digitalRead(button2)){//key 2 is pressed
36.     delay(200);
```

```
37. while(!digitalRead(button2));//wait for button release
38. MP3.nextTrack();//play the next song
39. }
40. }
```

### 3. Popis instrukcí

**MY1690.h:** Hlavní soubor modulu MP3 přehrávače.

**my1690 MP3:** Vytvoří objekt MP3.

**MP3.setLoopMode():** Nastaví modul MP3 přehrávače na přehrávání v smyčce.

**MP3.play():** Nastaví modul MP3 přehrávače na přehrávání skladeb.

**MP3.playPause():** Tento příkaz pozastaví přehrávání skladby, pokud je MP3 přehrávač v režimu přehrávání, a spustí přehrávání skladby, pokud je MP3 přehrávač v režimu pozastavení.

**MP3.previousTrack():** Ovládá MP3 přehrávač tak, aby přehrál předchozí skladbu.

**MP3.nextTrack():** Ovládá MP3 přehrávač tak, aby přehrál následující skladbu.



**!:** Logická operace NOT, když! Jakákoli následující hodnota je true nebo 1, výsledek je false nebo 0. Když! Když je další hodnota false nebo 0, výsledek je true nebo 1.

**while():** Smyčkové instrukce. Smyčka while se provádí, dokud není výraz v () nepravdivý, v tom okamžiku se smyčka ukončí. Syntaxe je následující:

```
while(Podmíněný výraz){  
Bloky kódu;  
}
```

Pokud je hodnota výrazu true nebo 1, blok kódu se opakovaně provádí, a pokud je hodnota výrazu false nebo 0, smyčka se ukončí.








## Oddíl 5 Hudební světlo (I)



### Cíl učebního plánu



-  Pochopení principu fungování světelného pásu RGB
-  Ovládněte používání světelného pásu RGB
-  Použití pásky RGB k dosažení chladných světelných efektů



**Viděli jste někdy světelné efekty na jevišti?  
Jaké vlastnosti podle vás mají?**



Hudba i světlo vyživují naše duše svým vlastním jedinečným způsobem. Hudba svou melodií a harmonií přináší k našim uším krásné melodie a uvolňuje a zpříjemňuje naše srdce; a světlo svým světlem a stínem,

transformace barev a světla, prostřednictvím vize vytvořit pro nás specifickou atmosféru, aby naše nálada mohla být šťastná a uvolněná.

Pokud se obojí spojí, hudba a světlo spolu spolupracují, můžeme si vychutnat nádhernou audiovizuální hostinu. V jevištním představení můžeme často vidět integraci hudby a osvětlení,





osvětlení se mění v rytmu hudby, takže si diváci mohou vychutnat krásu hudby nejen sluchem, ale také mohou zažít kouzlo osvětlení zrakem, více se ponořit do atmosféry vytvořené představením, aby se zlepšil divácký zážitek.

V této lekci vytvoříme chytrou hudební lampu pro chytrou domácnost. Při přehrávání hudby se s ní bude měnit i světlo, aby se zvýšil zážitek ze zábavy členů rodiny. Ačkoli jsme v předchozím kurzu používali světla RGB, jediná lampka RGB nemůže dosáhnout bohatých vizuálních efektů, takže v této lekci použijeme světelný pás RGB k dosažení světelných efektů.

### **Světelný pásek RGB**

RGB světelný pásek se skládá z červené, zelené a modré barvy LED světelných kuliček složených ze světelného pásku, díky různým barevným směsím lze dosáhnout různých barevných efektů.

V tomto kurzu používáme modul světelného pásku WS2812, který se skládá z několika propojených světel RGB LED. Každá LED dioda RGB je vybavena nezávislým řídicím čipem. Tyto řídicí čipy přijímají a odesílají data sériovým způsobem, aby mohly nezávisle řídit jas a barvu každé světelné kuličky na pásku. V systému Arduino máme k dispozici celou řadu knihoven pro světelné pásy, což nám umožňuje pohodlnější ovládání světelného pásku LED. Na adrese





této lekci použijeme knihovnu Adafruit\_NeoPixel k ovládání světelného pásku RGB.



**DO:** Připojte I/O pin vývojové desky.

**+5V:** Připojte k 5V pinu řídicí desky.

**GND:** Připojte ke kolíku GND řídicí desky.



## Smart Home Lab

### Barevné světlo



#### Popis projektu

Každá světelná perla na RGB světelném pásku postupně mění jas a barvu, čímž vytváří efekt tekoucí vody.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name            | Quantity |
|---|-----------------|----------|
|  | ESP32 Board     | 1        |
|  | RGB Light Strip | 1        |
|  | USB Data Cable  | 1        |



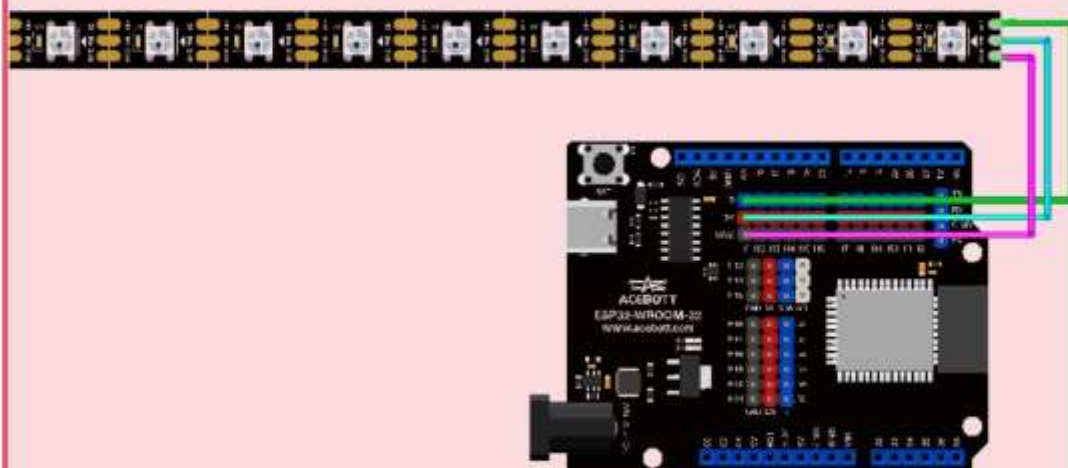


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 10 "10 Barevné světlo" montážního dokumentu.

## 3. Hardware Wiring Diagram





## Programmin

### 1. Program Flowchart



### 2. Coding

```
1. #include <Adafruit_NeoPixel.h> //importujte knihovnu pro ovládání pásku
2.
3. #define LED_PIN 5 //deklarování pinů RGB pásku
4. #define LED_COUNT 10 //deklarovat počet světel v RGB pásu
5. #define BRIGHTNESS 50 //nastavení jasu světla v rozsahu 0-255
6.
7. //pro vytvoření objektu proužku je třeba zadat počet korálků a číslo kolíku
   proužku.
8. Adafruit_NeoPixel strip(LED_COUNT, LED_PIN);
9.
10. void setup() {
11.     strip.begin(); //inicializace pásu
```

```

12. strip.setBrightness(BRIGHTNESS); //set the strip brightness
13. }
14.
15. void loop() {
16.   for(int i=0; i<LED_COUNT; i++) { //control each bead in turn
17.     //set the color of the beads to work to red:
18.     strip.setPixelColor(i, strip.Color(255, 0, 0));
19.     strip.show(); //update the status of the beads on the strip
20.     delay(30); //delay 30ms
21.   }
22.   for(int i=0; i<LED_COUNT; i++) { //control each bead in turn
23.     //set the color of the beads to work to green:
24.     strip.setPixelColor(i, strip.Color(0, 255, 0));
25.     strip.show(); //update the status of the beads on the strip
26.     delay(30); //delay 30ms
27.   }
28.   for(int i=0; i<LED_COUNT; i++) { //control each bead in turn
29.     //set the color of the beads to work to blue:
30.     strip.setPixelColor(i, strip.Color(0, 0, 255));
31.     strip.show(); //update the status of the beads on the strip
32.     delay(30); //delay 30ms
33.   }
34. }

```

### 3. Popis instrukcí

**Adafruit NeoPixel.h:** Knihovna pro ovládání RGB světelných pásků, která obsahuje řadu funkcí pro ovládání RGB světelných pásků a související data.

**Adafruit\_NeoPixel strip(count,pin):** Vytvoří objekt strip. „strip“ je název objektu a lze jej přizpůsobit. „count“ je počet LED diod v LED pásku a „pin“ je číslo LED pinu pásku.



**strip.begin():** Inicializuje pás.

**strip.setBrightness():** Nastaví jas pásu. Rozsah jasu je 0-255.

**strip.Color(R,G,B):** Nastaví hodnotu barvy korálku, každý parametr v rozsahu 0-255.

**strip.setPixelColor(i,color):** Nastaví barvu určené LED perly, kde „i“ je číslo perly a „color“ je hodnota barvy, která se má rozsvítit, získaná z strip.Color().

**strip.show():** Aktualizuje stav každé perly na pásku.



### Optimalizace schématu

Zesilte svítivost LEDek, například tak, že každá LEDka bude svítit jinou barvou a blikat.






## Oddíl 6 Hudební světlo (II)



### Cíl učebního plánu



-  Pochopení fungování zvukových senzorů
-  Zvládnutí používání zvukových senzorů
-  Použití světelného pásu RGB, přehrávače MP3, zvukového senzoru k dosažení funkce hudební lampy



### Jak může světlo rozumět hudbě?



Aby bylo možné dosáhnout efektu změny světla v závislosti na hudbě, je nutné zajistit, aby světelné zařízení mohlo přijímat hudební signál. Změna hudby je v podstatě změnou zvuku, proto můžeme k detekci změn hudby použít zvukové senzory. Vhodným propojením a nastavením zvukového snímače, modulu pro přehrávání hudby a osvětlovacího zařízení je možné řídit změny jasu a barev světla podle rytmu nebo hlasitosti hudby, a vytvořit tak rozmanitější a barevnější vizuální efekt.



## Snímač zvuku

Snímač zvuku je elektronické zařízení speciálně navržené pro snímání zvukových vln, známé také jako detektor zvuku nebo zvukový snímač. Jeho hlavní funkcí je převádět zvuky z



na zvuk. okolního prostředí na elektrické signály pro následné zpracování a analýzu, například měření intenzity, frekvence, tvaru vlny a dalších informací o zvuku. Tento typ snímače má široké využití v mnoha oblastech, včetně rozpoznávání hlasu, monitorování hluku v prostředí a zpracování zvukových signálů.

V tomto kurzu budeme používat modul zvukového senzoru pro hlasitosti zvukových signálů v okolním prostředí. Tento modul se skládá především z mikrofonu, napájecího zdroje a příslušných elektronických součástek. Mikrofon je kapacitního typu. Zvukové vlny v okolí způsobují rozkmitání membrány uvnitř mikrofonu, což vede změně kapacity, která následně vytváří odpovídající napětí.



změnu, čímž se dosáhne převodu zvukových signálů na elektrické signály. Když modul detekuje zvuk, odráží změny amplitudy zvuků okolního prostředí prostřednictvím výstupu analogového signálu. Čím větší je hodnota, tím větší je amplituda zvuku, což také znamená, že zvuk je hlasitější.



**S:** Připojte k analogovému I/O pinu řídicí desky.

**V:** Připojte k 5V pinu řídicí desky.

**G:** Připojte ke kolíku GND řídicí jednotky.



### Tips

V tomto kurzu se vzhledem k rozložení pinů vývojové desky ESP32 doporučuje připojit signální pin zvukového senzoru k jednomu z následujících pinů: 12, 13, 14, 15, 25, 26 nebo 17.





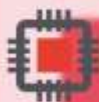
## Smart Home Lab

### Hudební světlo



#### Popis projektu

Vytvořte hudební světlo a nechte jeho jas měnit se podle hudby.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name              | Quantity |
|---|-------------------|----------|
|  | ESP32 Board       | 1        |
|  | RGB Light Strip   | 1        |
|  | Sound Sensor      | 1        |
|  | MP3 Player Module | 1        |
|  | Button Module     | 2        |
|  | Touch Sensor      | 1        |
|  | USB Data Cable    | 1        |



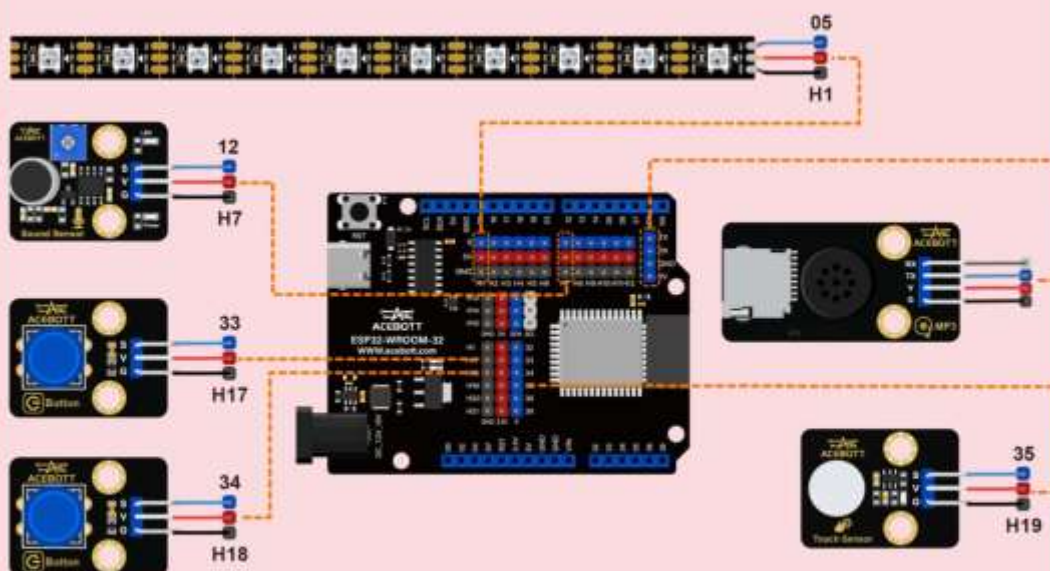


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 11 "11 Hudební světlo" montážního dokumentu.

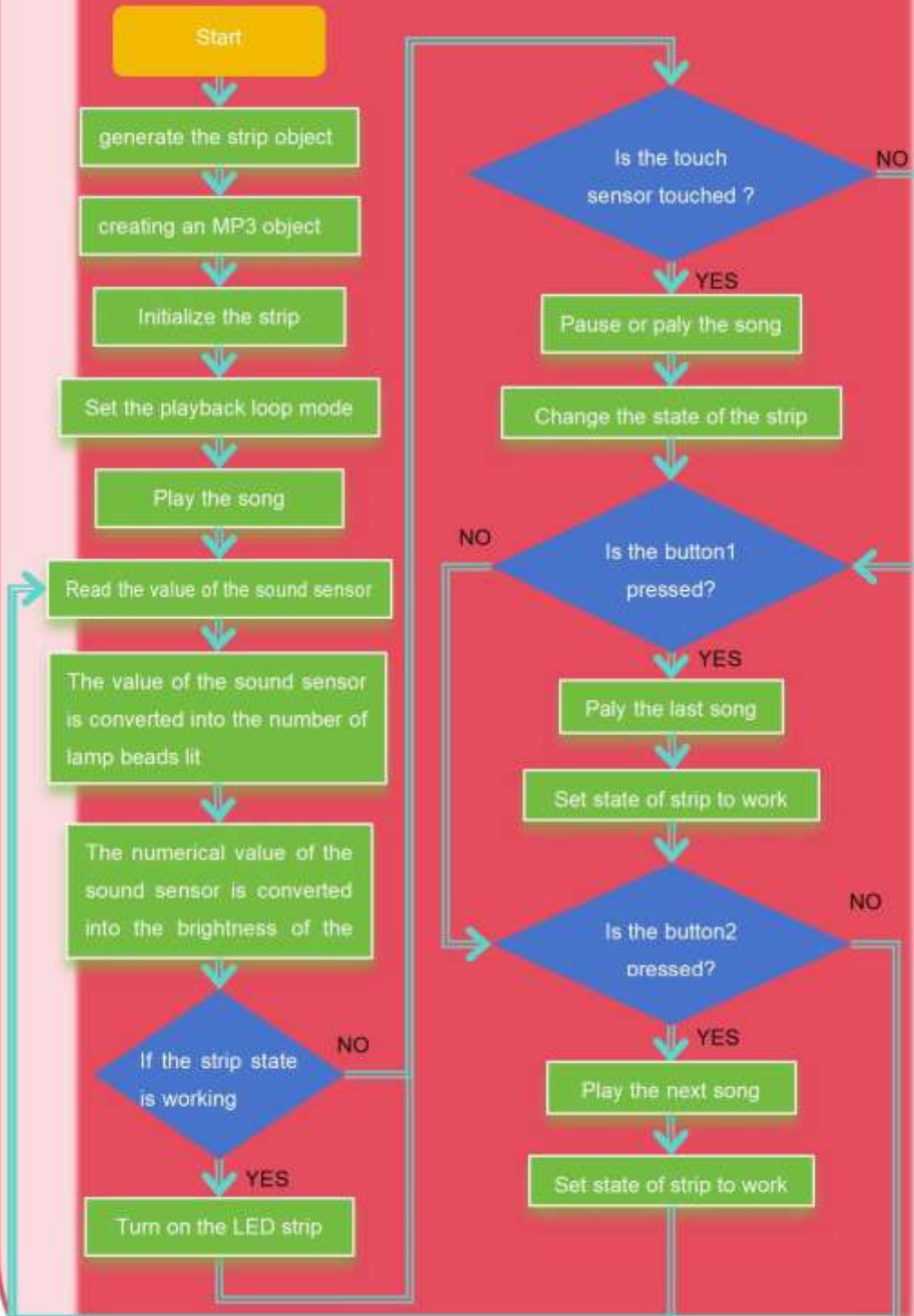
## 3. Hardware Wiring Diagram





## 编程实现

### 1. 程序流程图



## 2.Coding

```
1. #include <Adafruit_NeoPixel.h> //import strip Library
2. #include <MY1690.h> //import mp3 Library
3.
4. #define LED_PIN 5 //declare the pin of LED strip
5. #define sound 12 //declare the pin of sound sensor
6. #define button1 33 //declare the pin of button1
7. #define button2 34 //declare the pin of button2
8. #define touch 35 //declare the pin of touch sensor
9. #define LED_COUNT 10 //declare the pin of button1
10.
11. //creat an object of strip.
12. Adafruit_NeoPixel strip(LED_COUNT, LED_PIN);
13. my1690 MP3; //creat an object of MP3
14. //initialize the state of strip,
15. //'true' indicates strip is working,
16. //'false' indicates strip is not working:
17. bool strip_state = true;
18.
19. void setup() {
20.   strip.begin(); // initialize the LED strip
21.   // set the pin of sound sensor as input:
22.   pinMode(sound, INPUT);
23.   // set the pin of touch sensor as input:
24.   pinMode(touch, INPUT);
25.   pinMode(button1, INPUT); // set the pin of button1 as input.
26.   pinMode(button2, INPUT); // set the pin of button2 as input.
27.   Serial.begin(9600); //set baud rate to 9600.
28.   delay(100); // wait 100ms
29.   MP3.setLoopMode(); //set the MP3 to Loop playback mode.
30.   MP3.play(); //play the song
31. }
32.
33. //define the function that realize the music light,
34. //color is the parameter which control colors of LED strip
35. void MusicLight(uint32_t color){
36.   //get the soundness from the sound sensor
```

```

37. int soundness = analogRead(sound);
38. Serial.println(soundness);
39. /*convert the detected sound intensity into the number
40. of LED beads to be illuminated*/
41. int LED_num = map(soundness,1800,3500,0,10);
42. /*convert the detected sound intensity into the
43. brightness of LED beads*/
44. int brightness = map(soundness,1800,4095,0,200);
45. //set the brightness of LED strip
46. strip.setBrightness(brightness);
47. if(strip_state){
48. // control every bead on LED strip in turn
49. for(int i=0; i<LED_num; i++) {
50. //set the color of the LED beads
51. strip.setPixelColor(i, color);
52. strip.show(); //display the color of the LED bead
53. delay(30); // wait for 30ms
54. }
55. }
56. strip.clear();//shut down the strip
57. if(!digitalRead(touch)){//if the touch sensor is touched
58. delay(200);// wait 0.2 seconds
59. //wait the hand to leave the touch sensor:
60. while(!digitalRead(touch));
61. /*If the MP3 is currently playing, then pause playback;
62. if it is currently paused, then play.*/
63. MP3.playPause();
64. strip_state = !strip_state;//change the state of strip
65. }
66. if(!digitalRead(button1)){//if the button1 is pressed
67. delay(200);// wait 0.2 seconds
68. while(!digitalRead(button1));//waitbutton1 to released
69. MP3.previousTrack();//play the last song
70. strip_state = true;//set the state of strip to true
71. }
72. if(!digitalRead(button2)){//if the button2 is pressed

```

```

73.   delay(200); // wait 0.2 seconds
74.   while(!digitalRead(button2)); //wait button2 to released
75.   MP3.nextTrack(); //play the next song
76.   strip_state = true; //set the state of strip to true
77. }
78. }
79. void loop() {
80.   //set the color of music light as red:
81.   MusicLight(strip.Color(255,0,0));
82.   //set the color of music light as green:
83.   MusicLight(strip.Color(0,255,0));
84.   //set the color of music light as blue:
85.   MusicLight(strip.Color(0,0,255));
86.   //set the color of music light as yellow:
87.   MusicLight(strip.Color(255,255,0));
88.   //set the color of music light as purple:
89.   MusicLight(strip.Color(255,0,255));
90.   //set the color of music light as cyan:
91.   MusicLight(strip.Color(0,255,255));
92.   //set the color of music light as white:
93.   MusicLight(strip.Color(255,255,255));
94. }

```

### 3. Popis příkazů

#### **map(value,fromHigh,fromLow,toHigh,toLow):**

Mapuje hodnotu proporcionálně z jednoho rozsahu do druhého. Zde „value“ je hodnota, která má být mapována, „fromHigh“ je minimální hodnota původního rozsahu, „fromLow“ je maximální hodnota původního rozsahu, „toHigh“ je minimální hodnota cílového rozsahu a „toLow“ je maximální hodnota cílového rozsahu.

**strip.clear():** Příkaz pro vymazání obrazovky světelných proužků, který vypne všechna světla.

## Kapitola 4 Teplý a pohodlný domov

Dobré životní prostředí má významný vliv na kvalitu a pohodlí rodinného života. V následujících kapitolách se budeme zabývat tím, jak zkonstruovat inteligentní systém monitorování prostředí v inteligentní domácnosti, systém schopný automaticky zjišťovat klíčové ukazatele, jako je teplota, vlhkost a kvalita vzduchu v místnosti. Prostřednictvím samoregulační funkce systému Smart Home můžeme zajistit, aby obytné prostředí zůstalo vhodné a pohodlné po všechna čtyři roční období. Aplikace tohoto inteligentního systému monitorování prostředí nám vytvoří kvalitnější a pohodlnější domácí životní prostředí.



## Oddíl 1 Detekce teploty a vlhkosti



### Cíl učebního plánu

- ✎ pochopení vlivu teploty a vlhkosti na domácí prostředí
- ✎ Pochopení principu čidla teploty a vlhkosti
- ✎ Ovládejte používání čidla teploty a vlhkosti



**Víte, jak měřit teplotu a vlhkost? Jak teplota a vlhkost ovlivňují náš život?**

Vzhledem k sezónním změnám se teplota a vlhkost uvnitř a vně místnosti odpovídajícím způsobem mění. V chladné zimě je třeba udržovat teplo v interiéru, zatímco v horkém létě je třeba udržovat teplo v interiéru.



v interiéru je nutné chladit. Vlhké počasí může vést ke zvýšené vlhkosti v interiéru. Pokud nelze teplotu a vlhkost v interiéru



regulované, může vést k fyzickému nepohodlí a ovlivnit kvalitu života. Nevhodná teplota a vlhkost mohou mít dokonce vliv na naše zdraví, například prostředí s vysokou teplotou a vlhkostí může způsobit úpal, zatímco prostředí s nízkou teplotou a vlhkostí může vést k suché pokožce a dýchacím potížím. Udržování vhodné vnitřní teploty a vlhkosti je proto zásadní. V systému inteligentní domácnosti můžeme vytvořit automatický systém řízení teploty a vlhkosti za předpokladu, že vnitřní teplota a vlhkost jsou monitorovány v reálném čase.

Při tradičních metodách měření se k měření teploty a vlhkosti obvykle používají mechanická zařízení, jako jsou teploměry a vlhkoměry, ale tento přístup často zahrnuje značné chyby. Pro zvýšení přesnosti měření představíme v tomto kurzu metodu použití čidel teploty a vlhkosti pro sledování teploty a vlhkosti v interiéru.

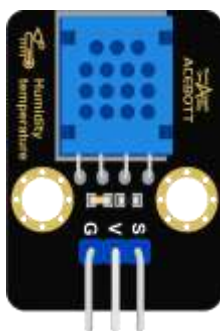
## Snímač teploty a vlhkosti

Snímač teploty a vlhkosti je zařízení, které se používá k měření teploty a vlhkosti prostředí. Typem čidla teploty a vlhkosti použitým v tomto kurzu je DHT11, které měří teplotu a vlhkost v prostředí.





prostředí prostřednictvím vnitřních elektronických součástí. Pro měření teploty se používá součástka známá jako termistor, zatímco měření vlhkosti se provádí pomocí kondenzátoru citlivého na vlhkost. DHT11 vystupuje z výsledků měření digitálně, což znamená, že jej lze přímo připojit k řídicí desce a přenášet data prostřednictvím digitálního rozhraní. DHT11 se široce používá v různých aplikacích, jako je monitorování vnitřní teploty a vlhkosti, meteorologické stanice, řízení skleníků, systémy inteligentních domácností atd., zejména ve scénářích, které vyžadují digitální údaje o teplotě a vlhkosti v reálném čase.



**S:** Připojte digitální I/O pin řídicí desky.

**V:** Připojte 5V pin řídicí desky.

**G:** Připojte kolík GND řídicí desky.





## Smart Home Lab



### Detekce T & H





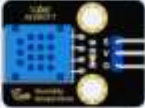

#### Popis projektu

Aktuální hodnota teploty a vlhkosti se zobrazuje v reálném čase na digitálním displeji.



#### Hardwarové schéma

1 .Seznam hardwaru

| Picture   | name                 | Quantity |
|---|----------------------|----------|
|  | ESP32 Board          | 1        |
|  | 4-Digit Tube Display | 1        |
|  | DHT11                | 1        |
|  | USB Data Cable       | 1        |



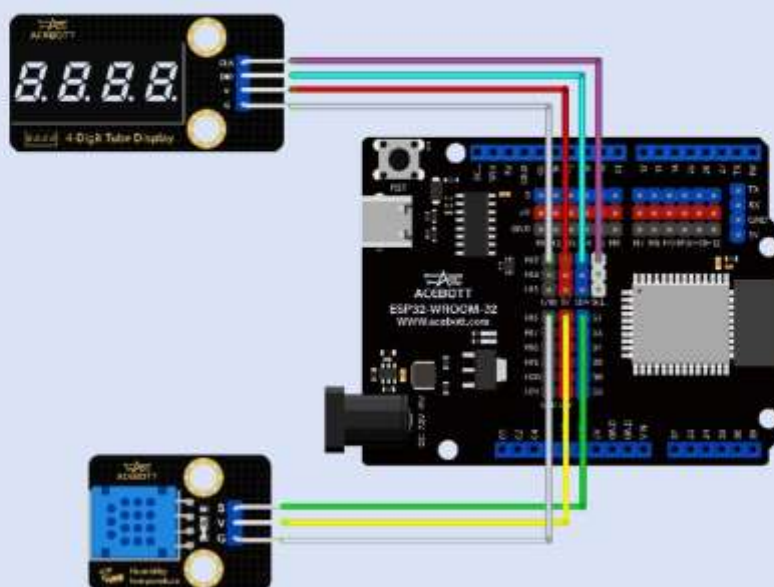


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 12 "12 Inteligentní detekce T&H" montážního dokumentu.

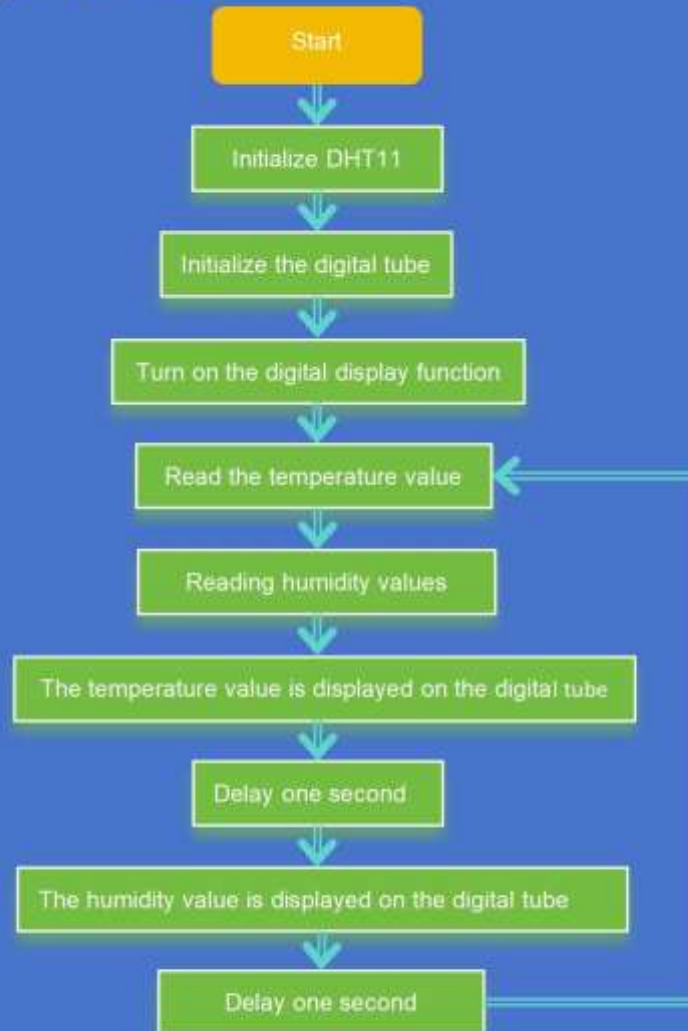
## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. #include <DHT.h> //import the DHT sensor library
2. #include <Wire.h> //introduce the I2C communication library
3. #include <TM1650.h> //import digital tube Library
4.
5. #define DHTPIN 32 //declare the signal pin of the DHT sensor
6. #define DHTTYPE DHT11 //declare the DHT sensor model
```

```

7. char dis[5]; //create a character array to hold the displayed string
8. DHT dht(DHTPIN,DHTTYPE); //create a DHT object
9. TM1650 d; //create a digital object
10.
11. void setup(){
12.   dht.begin(); //initialize the DHT sensor
13.   Wire.begin(); //enable I2C communication
14.   d.init(); //initialize the digital tube object
15.   d.setBrightness(TM1650_MIN_BRIGHT); //Set the digital tube brightness
16.   d.displayOn(); //enable the digital display function
17. }
18.
19. void loop() {
20.   float h = dht.readHumidity(); //read humidity value
21.   float t = dht.readTemperature(); //get the temperature (Celsius)
22.   itoa(h*100,dis,10); //convert humidity values to a character array
23.   Serial.println(dis);
24.   delay(1000);
25.   d.displayString(dis); //display humidity on the digital tube
26.   d.setDot(1,true); //set the decimal position of the digital tube
27.   delay(1000);
28.   itoa(t*100,dis,10); //convert humidity values to a character array
29.   d.displayString(dis); //display the temperature on the digital tube
30.   d.setDot(1,true); //set the decimal position of the digital tube
31.   delay(1000);
32. }

```

### 3. Popis příkazů

**DHT dht(pin,dhtType):** Tento příkaz vytvoří objekt pro senzor DHT, který obsahuje různé metody pro ovládání senzoru DHT. Zde „pin“ je číslo pinu, ke kterému je připojen signální port senzoru DHT, a „dhtType“ je model senzoru DHT.

**dht.readHumidity():** Čte hodnotu vlhkosti.

**dht.readTemperature():** Čte hodnotu teploty.

## Oddíl 2 Chytrý ventilátor



### Cíl učebního plánu

- ✎ pochopení vlivu teploty na domácí prostředí
- ✎ Pochopení principu stejnosměrného motoru
- ✎ Zvládnutí používání stejnosměrného motoru
- ✎ Použití snímače teploty a vlhkosti a stejnosměrného motoru k vytvoření inteligentního systému ventilátoru



**Máte rádi léto nebo zimu? Jak se doma ochlazujete a zahříváte?**

Rozsah teplot, které lidské tělo snese, se u jednotlivých osob liší v závislosti na mnoha faktorech, jako je vlhkost vzduchu, rychlost větru, oblečení, příjem vody, zdravotní stav atd.



Obecně platí, že když teplota přesáhne  $37,5\text{ }^{\circ}\text{C}$ , lidské tělo se začne cítit nepříjemně kvůli horku; když se teplota zvýší nad  $37,5\text{ }^{\circ}\text{C}$ , lidské tělo se začne cítit nepříjemně.



teplota klesne pod 15 °C, tělo začne ztrácet teplo a potřebuje další zahřívací opatření. Když teplota našeho životního prostředí překročí rozsah, který jsme schopni tolerovat, přijmeme opatření k ochlazení nebo k udržení tepla, aby se nám žilo pohodlněji. Například v prostředí s vysokou teplotou lze ke snížení vnitřní teploty použít ventilátory nebo klimatizaci. V chladné zimě mohou lidé zvýšení vnitřní teploty použít topení, elektrické deky nebo silnější přikrývky.

Víte, jak se ventilátor ochlazuje?



Princípem ochlazování pomocí ventilátoru je rychlá cirkulace vzduchu, která podporuje odpařování mezi horkým vzduchem a vlhkou pokožkou na povrchu těla, a tím odvádí teplo. Když ventilátor fouká na vlhkou pokožku,

odpařování vody vyžaduje teplo, takže okolní teplota je nižší. Ventilátor navíc zrychluje pohyb vzduchu, odvádí horký vzduch kolem těla a umožňuje, aby ho nahradil čerstvý, chladný vzduch, čímž se dosáhne chladicího účinku. Díky tomuto procesu se tělo cítí chladnější, zejména v horkém prostředí, kde se ventilátor stává jednoduchým a účinným chladicím nástrojem.

V tomto kurzu budeme budovat chytrý ventilátorový systém , který .





automaticky aktivuje ventilátor k ochlazení, když teplota dosáhne nastavené hodnoty. Ventilátor se skládá především z motoru a lopatek a funguje tak, že řídí otáčení motoru, který pohání lopatky. Proto, abychom mohli ventilátor řídit, musíme pochopit, jak řídit činnost motoru.

## Motor

Motor je zařízení, které přeměňuje elektrickou energii na mechanickou. V podstatě se skládá ze dvou hlavních částí: statoru a rotoru. Stator je



stacionární část motoru, obvykle tvořená vinutím (cívkami) a železným jádrem. Magnetické pole vytvářené ve vinutí elektrickým proudem se nazývá statorové pole. Rotor je pohyblivá část motoru, obvykle tvořená vodiči. Když rotorem prochází proud, působí na něj síla ve statorovém poli, čímž vzniká mechanický pohyb, který způsobuje otáčení rotoru. To je základní princip činnosti motoru.

Motory lze rozdělit na různé typy podle jejich konstrukce, včetně motorů na stejnosměrný proud (DC), motorů na střídavý proud (AC),



krokové motory, servomotory atd. V tomto článku se zaměříme na stejnosměrné motory. Řízení stejnosměrných motorů je poměrně jednoduché. Obecně,





stejnoseměrný motor má dvě svorky: jednu připojenou ke kladnému pólu zdroje a druhou k zápornému pólu. Tímto způsobem se motor uvede do chodu. Pokud kladný a záporný pól prohodí, motor se bude pohybovat opačným směrem.



### Tips

Provoz motoru vyžaduje určité napětí a proud. Vzhledem k tomu, že výstupní proud portů I/O vývojové desky je velmi malý a nestačí k pohonu stejnosměrného motoru, nelze piny stejnosměrného motoru připojit přímo k portům I/O vývojové desky. K řízení provozu stejnosměrného motoru prostřednictvím portů I/O řídicí desky je zapotřebí obvod ovladače motoru.

V tomto kurzu používáme modul stejnosměrného motoru, který se skládá ze stejnosměrného motoru, obvodu pohonu motoru, napájecího zdroje a dalších částí. Dokáže řídit směr a rychlost pohybu motoru. Modul má dva signální kolíky IN- a IN+, respektive připojené k I/O portu řídicí desky, když jsou tyto dva kolíky na vstupu vysoké nebo nízké úrovně, motor nepracuje; Motor pracuje, když jeden ze dvou kolíků na vstupu má vysokou úroveň a jeden nízkou úroveň.

**IN-:** Připojte I/O pin řídicí desky.

**IN+:** Připojte I/O pin řídicí desky.

**VCC:** Připojte 5V pin řídicí desky.

**GND:** Připojte pin GND řídicí jednotky.





## Smart Home Lab



### Inteligentní ventilátor



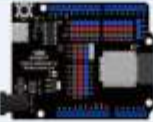



#### Popis projektu

Když vnitřní teplota překročí nastavenou hodnotu, ventilátor se automaticky spustí; v opačném případě ventilátor nefunguje.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name            | Quantity |
|---|-----------------|----------|
|  | ESP32 Board     | 1        |
|  | DC Motor Module | 1        |
|  | DHT11           | 1        |
|  | USB Data Cable  | 1        |



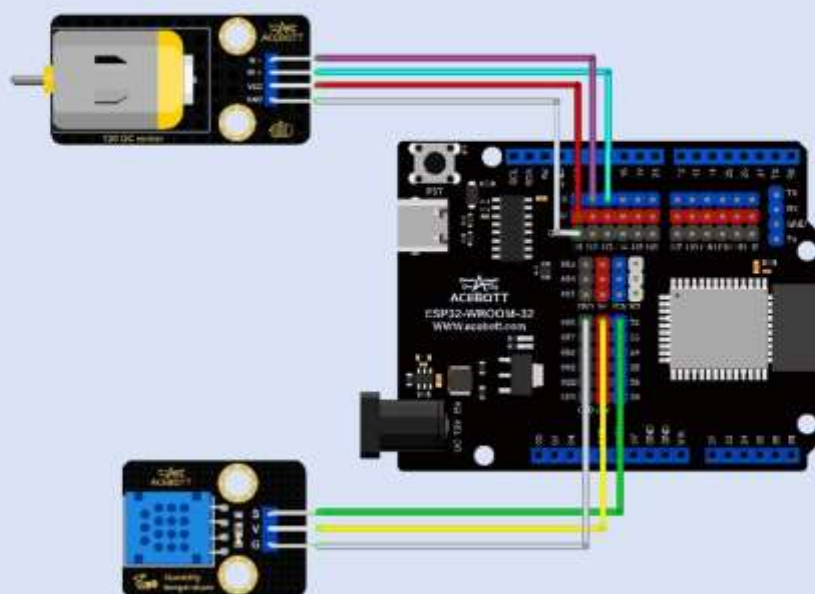


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 13 "13 Inteligentní ventilátor" montážního dokumentu.

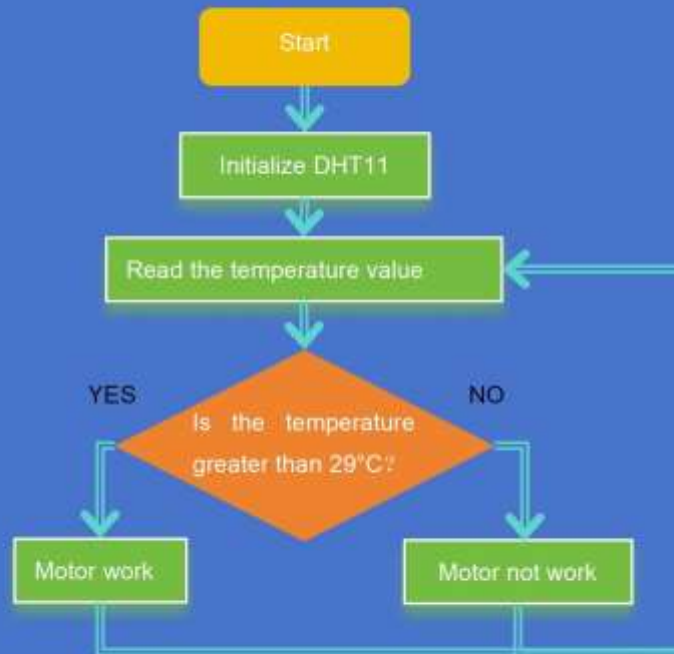
## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



### 2. Coding

```
1. #include <DHT.h>
2.
3. #define DHTPIN 32 //declare the signal pin of the DHT sensor
4. #define DHTTYPE DHT11 //declare the DHT sensor model
5. #define IN1 16 //declare pin 1 of DC motor
6. #define IN2 17 //declare pin 2 of DC motor
7.
8. DHT dht(DHTPIN, DHTTYPE); //create a DHT object
9.
10. void setup() {
11. dht.begin(); //initialize the DHT sensor
12. pinMode(IN1,OUTPUT); //set motor pin 1 to output mode
13. pinMode(IN2,OUTPUT); //set motor pin 2 to output mode
```

```

14. Serial.begin(115200);
15. }
16.
17. void loop() {
18.   //read the temperature, default is Celsius:
19.   float t = dht.readTemperature();
20.   Serial.println(t);
21.   //check whether the temperature is greater than 29°C :
22.   if (t>29){
23.     //motor work:
24.     digitalWrite(IN1,HIGH);
25.     digitalWrite(IN2,LOW);
26.   }
27.   else{
28.     //motor not working:
29.     digitalWrite(IN1,HIGH);
30.     digitalWrite(IN2,HIGH);
31.   }
32. }

```



### Optimalizace schématu

Přidejte do inteligentního ventilátoru funkci automatického nastavení, která nastaví naměřenou teplotu na tři úrovně, přičemž každá úroveň odpovídá konkrétní rychlosti ventilátoru. Čím vyšší je teplota, tím rychleji se ventilátor otáčí.



### Tips

Rychlost motoru lze řídit pomocí signálu PWM. Konkrétní úprava kódu spočívá ve změně výstupu pinu s vysokou úrovní na výstup PWM. Odkaz na kód je následující:

```

analogWrite(IN1 ,speed);//rozsah rychlosti: 0-255
digitalWrite(IN2,LOW);

```

## Oddíl 3 Regulace teploty a vlhkosti



### Cíl učebního plánu

- ✎ Pochopení vlivu vlhkosti na život v domácnosti
- ✎ Pochopení principu rozprašovače
- ✎ Ovládněte používání rozprašovače
- ✎ Inteligentní systém řízení teploty a vlhkosti je vytvořen pomocí teplotních a vlhkostních čidel, motorů a rozprašovačů.



**Znáte rozsah vlhkosti vhodný pro lidské bydlení?  
Jaké jsou některé způsoby odvlhčování a  
zvlhčování?**

Vlhkost vzduchu vyjadřuje relativní obsah molekul vody ve vzduchu. Vlhkost vzduchu v našem každodenním životě významný vliv na pohodlí a zdraví člověka. Obecně platí, že nejvhodnější vlhkost prostředí pro lidský organismus je mezi 40% a 65% relativní vlhkost. V tomto rozmezí vlhkosti se





se lidské tělo cítí nejpohodlněji a je méně pravděpodobné, že se v něm budou šířit choroboplodné zárodky. Pokud je však vlhkost příliš nízká, může způsobit problémy, jako je suchá pokožka a nepříjemné pocity v krku; vysoká vlhkost může vést k růstu plísní a vlhkého oblečení. Pokud je vlhkost příliš vysoká, můžeme vlhkost v místnosti snížit pomocí odvlhčovačů, klimatizací a dalších elektrických spotřebičů nebo větráním a používáním vysoušečů. Když je vlhkost příliš nízká, můžeme vlhkost v místnosti zvýšit pomocí zvlhčovačů vzduchu, rozprašováním vodní mlhy nebo vytíráním podlahy a kropením vodou.

Je třeba si uvědomit, že různé oblasti, roční období a osobní preference mohou vyžadovat různé úrovně vlhkosti v interiéru. Proto musíme zvolit vhodné metody odvlhčování a zvlhčování podle aktuální situace.

Pro zvýšení komfortu chytrých domácností vybudujeme inteligentní systém regulace teploty a vlhkosti, který bude regulovat teplotu a vlhkost v interiéru. V tomto systému použijeme rozprašovač k dosažení zvlhčovací funkce.

### Rozprašovač

Rozprašovač je zařízení, které dokáže přeměnit kapalinu na drobné kapičky nebo mlhu. Podle principu činnosti se rozprašovače dělí na ultrazvukové rozprašovače, rozprašovače se stlačeným vzduchem, rozprašovací rozprašovače a tepelné rozprašovače.





Zde se zaměříme na ultrazvukový rozprašovač, který využívá ultrazvukové vibrace k rozptýlení kapaliny na malé kapičky. Zařízení obsahuje ultrazvukový oscilátor, který generuje vysokofrekvenční ultrazvukové vibrace. Kapalina působením ultrazvukových vln vytváří drobné kapičky a je vypuzena ve formě mlhy. Tento typ rozprašovače se široce používá v lékařských rozprašovačích, aroma difuzérech a dalších oblastech.

V tomto kurzu používáme modul zvlhčovače, který obsahuje rozprašovací desku, řídicí obvod, napájecí zdroj a související elektronické součástky. Modul je spouštěn klesající hranou impulsu. Když signální kolík rozprašovače detekuje signál s klesající hranou, rozprašovač začne stříkat. Pokud opět detekuje signál klesající hrany, rozprašovač přestane stříkat a tento proces se opakuje.



**S:** Připojte digitální I/O pin řídicí desky. Spouštění při klesající hraně.

**V:** Připojte 5V pin řídicí desky.

**G:** Připojte pin GND řídicí desky.

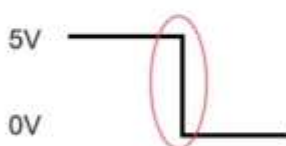




## Vysvětlení pojmů

### Klesající hrana (nebo také sestupná)

V digitálních obvodech se okamžik, kdy se digitální úroveň změní z vysoké úrovně (digitální „1“) na nízkou úroveň (digitální „0“), nazývá **klesající hrana**. Klesající hrany se často používají jako referenční body pro spouštěcí a synchronizační signály a lze je použít k řízení chování jiných součástí obvodu.



V Arduino IDE je referenční program pro generování klesající hrany následující:

```
digitalWrite (pin,HIGH);  
delay(500);  
digitalWrite(pin,LOW);
```



### Tips

Při použití modulu zvlhčovače je po nahrání programu nutné odpojit datový kabel a poté jej napájet z externího zdroje.





## Smart Home Lab

### Regulace teploty a vlhkosti








#### Popis projektu

Navrhněte inteligentní systém regulace teploty a vlhkosti s displejem pro zobrazení aktuální teploty a vlhkosti v místnosti. Pokud je teplota v místnosti příliš vysoká, automaticky se zapne ventilátor. Pokud je vlhkost v místnosti příliš nízká, automaticky se zapne zvlhčovač vzduchu.



#### Hardwarové schéma

##### 1. Seznam hardwaru

| Picture   | name                   | Quantity |
|---|------------------------|----------|
|  | ESP32 Board            | 1        |
|  | Humidifier Module      | 1        |
|  | DHT11                  | 1        |
|  | Dc motor module        | 1        |
|  | 4-Digit Tube Displayer | 1        |
|  | USB Data Cable         | 1        |



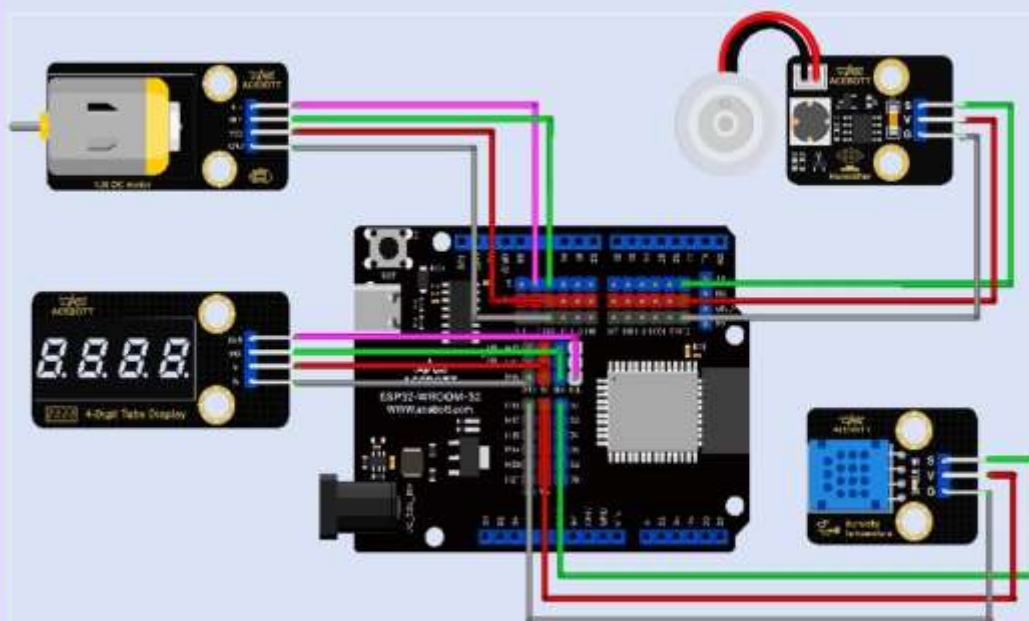


## 2. Hardware Structure Diagram



Podrobný postup montáže naleznete v části 14 "14 Inteligentní řídicí systém T&H" montážního dokumentu.

## 3. Hardware Wiring Diagram





## Programming

### 1. Program Flowchart



## 2.Coding

```
1. #include <DHT.h>
2. #include <Wire.h>
3. #include <TM1650.h>
4.
5.
6. #define DHTPIN 32 //declare the signal pin of the DHT sensor
7. #define DHTTYPE DHT11 //declare the DHT sensor model
8. #define IN1 16 //declare pin 1 of DC motor
9. #define IN2 17 //declare pin 2 of DC motor
10. #define atomizer 27 //declare the atomizer pin
11.
12.
13. char dis[5]; //create a character array to hold the displayed string
14. DHT dht(DHTPIN, DHTTYPE); //create a DHT object
15. TM1650 d; //create a digital object
16. //initialize the atomizer working state, false is off, true is on
17. bool atomizer_state = false ;
18.
19.
20. void setup() {
21.   Serial.begin(9600);
22.   pinMode(IN1,OUTPUT)//set motor pin 1 to output mode
23.   pinMode(IN2,OUTPUT); //set motor pin 2 to output mode
24.   pinMode(atomizer,OUTPUT); //set the atomizer pin to output mode
25.   dht.begin(); //initialize the DHT sensor
26.   Wire.begin(); //enable I2C communication
27.   d.init(); //initialize the digital tube object
28.   // Set the brightness of the digital display
29.   d.setBrightness(TM1650_MIN_BRIGHT);
30.   d.displayOn(); //enable the digital display function
31. }
32.
33. void loop() {
34.   float h = dht.readHumidity(); //read humidity value
35.   float t = dht.readTemperature(); // Read the temperature
```

```

36. Serial.println(h);
37. Serial.println(t);
38. auto_fan(t);//intelligent control fan
39. auto_Atomizer(h);//intelligent control atomizer
40. DHT_Display(h,t);//display the temperature and humidity on digital tube
41. }
42.
43. void DHT_Display(float h,float t){
44. //create temperature and humidity display function
45. itoa(h*100,dis,10);//convert humidity values to a character array
46. d.displayString(dis);//display humidity on the digital tube
47. d.setDot(1,true);//set the decimal place
48. delay(500);
49. itoa(t*100,dis,10);//convert humidity values to a character array
50. d.displayString(dis);//display the temperature on the digital tube
51. d.setDot(1,true);//set the decimal place
52. delay(500);
53. }
54.
55. void auto_fan(float t){ //create the fan control function
56. //check if the temperature is greater than 29 ℃ :
57. if (t>29){//temperature greater than 29 ° C:
58. //motor work:
59. Serial.println("Motor Start");
60. digitalWrite(IN1,HIGH);
61. digitalWrite(IN2,LOW);
62. }
63. else{//temperature less than 29 ℃ :
64. //motor not working:
65. Serial.println("Motor Stop");
66. digitalWrite(IN1,HIGH);
67. digitalWrite(IN2,HIGH);
68. }
69. }
70.

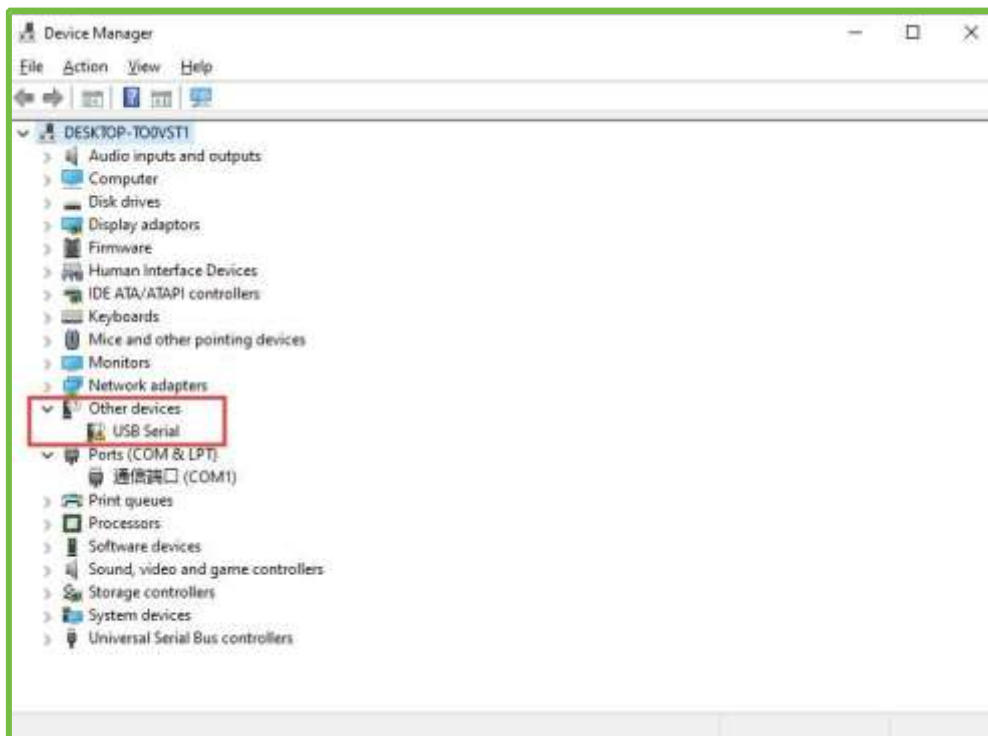
```

```
71. void auto_Atomizer(float h){//create atomizer control function
72.   if (h < 40){// Check if humidity is less than 40% :
73.     if(atomizer_state == false){//check if the atomizer is off
74.       atomizer_state = true ;//set the atomizer to on
75.       Serial.println("Atomizer Start");
76.       //falling edge triggers atomizer to work
77.       digitalWrite(atomizer,HIGH);
78.       delay(500);
79.       digitalWrite(atomizer,LOW);
80.     }
81.   }
82.   else{
83.     if(atomizer_state == true){//check if the atomizer is on
84.       atomizer_state = false ;//set the atomizer working state to off
85.       Serial.println("Atomizer Stop");
86.       //falling edge triggers atomizer shutdown
87.       digitalWrite(atomizer,HIGH);
88.       delay(500);
89.       digitalWrite(atomizer,LOW);
90.     }
91.   }
92. }
```

## ČASTO KLADENÉ DOTAZY

### Řešení selhání instalace ovladače

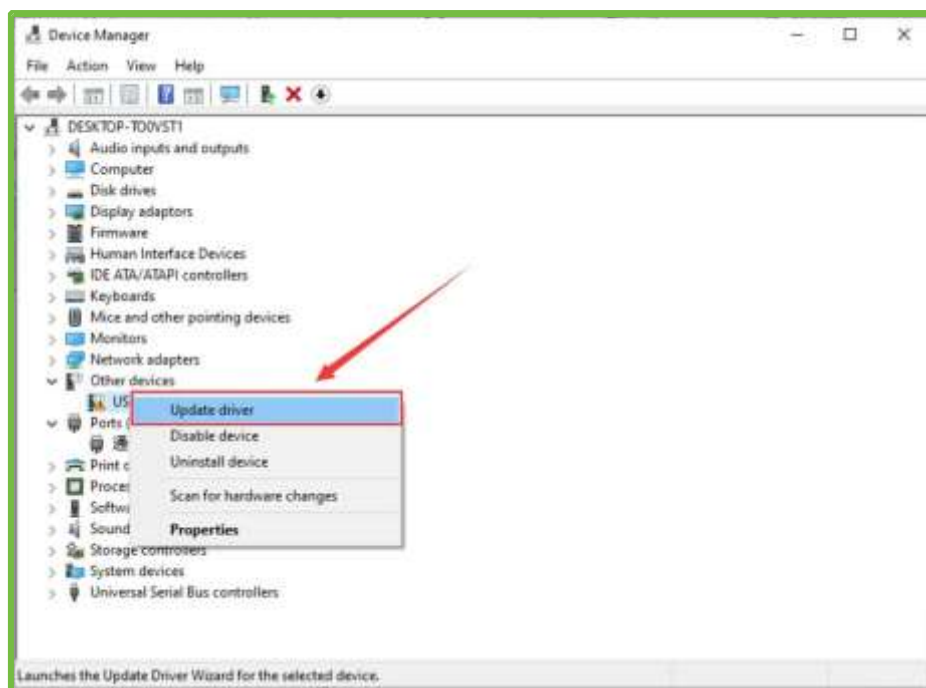
Pokud se instalace ovladače během procesu instalace Arduino IDE nezdaří, po prvním připojení řídicí desky ESP32 k počítači klikněte pravým tlačítkem myši na "Můj počítač" -> klikněte na "Vlastnosti" -> klikněte na "Správce zařízení". V části 'Ostatní zařízení' se zobrazí 'USB-Serial' nebo 'Neznámé zařízení'.



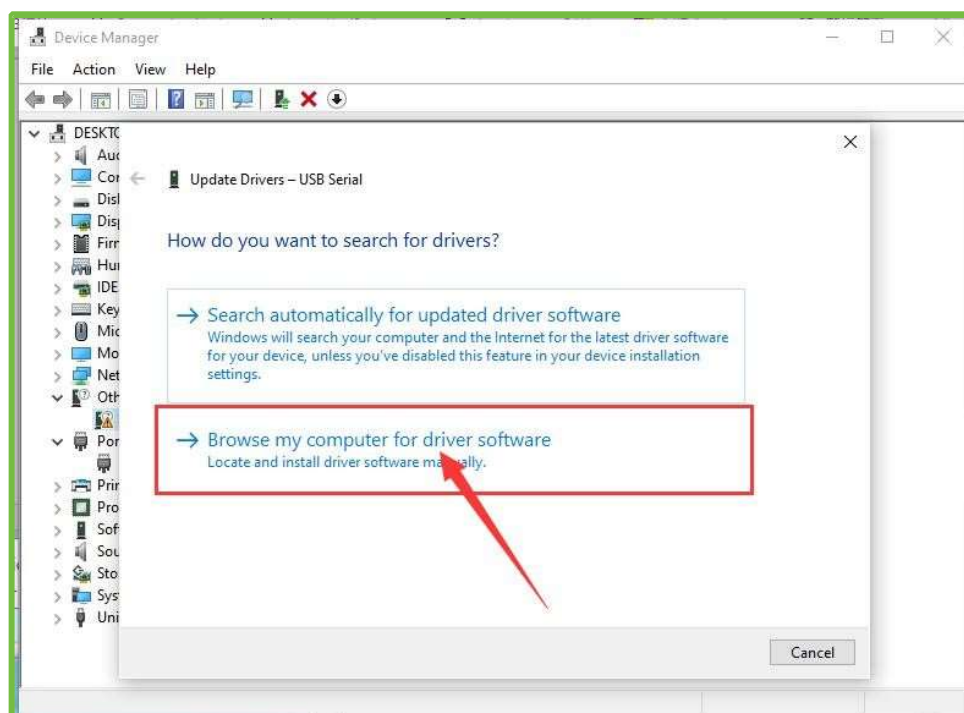
V tomto okamžiku je třeba mít místní soubor ovladače a ovladač nainstalovat ručně. Konkrétní kroky jsou následující:

Krok 1: Klikněte pravým tlačítkem myši na zařízení a vyberte možnost v horní nabídce (Aktualizace softwaru ovladače), jak je znázorněno na následujícím obrázku.



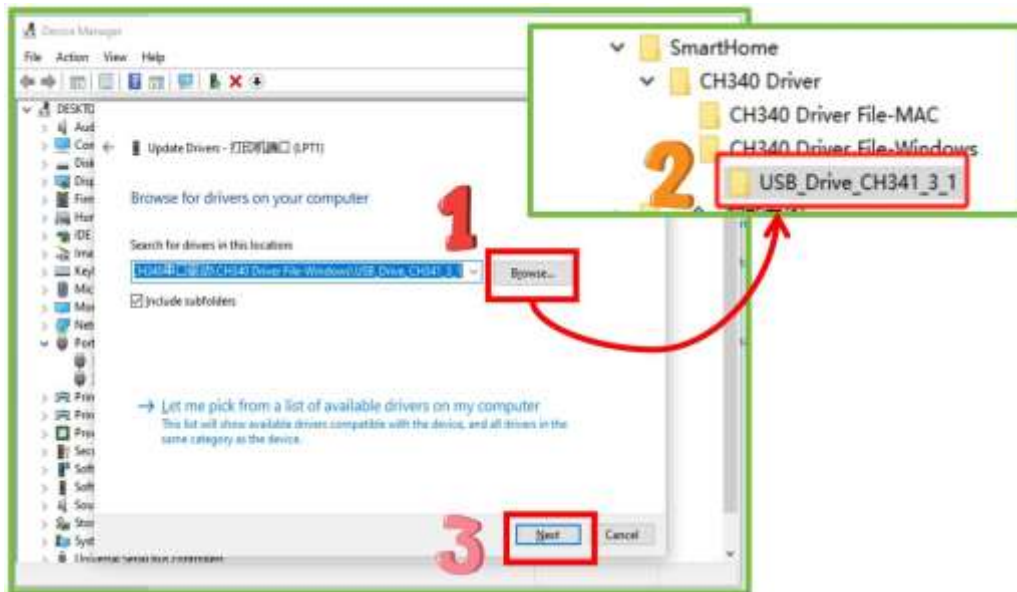


Krok 2: Poté se zobrazí výzva "Automaticky vyhledat aktualizovaný software ovladače" nebo "Procházet můj počítač pro software ovladače", jak je znázorněno na následujícím obrázku. Na této stránce vyberte možnost "Procházet můj počítač pro software ovladače".

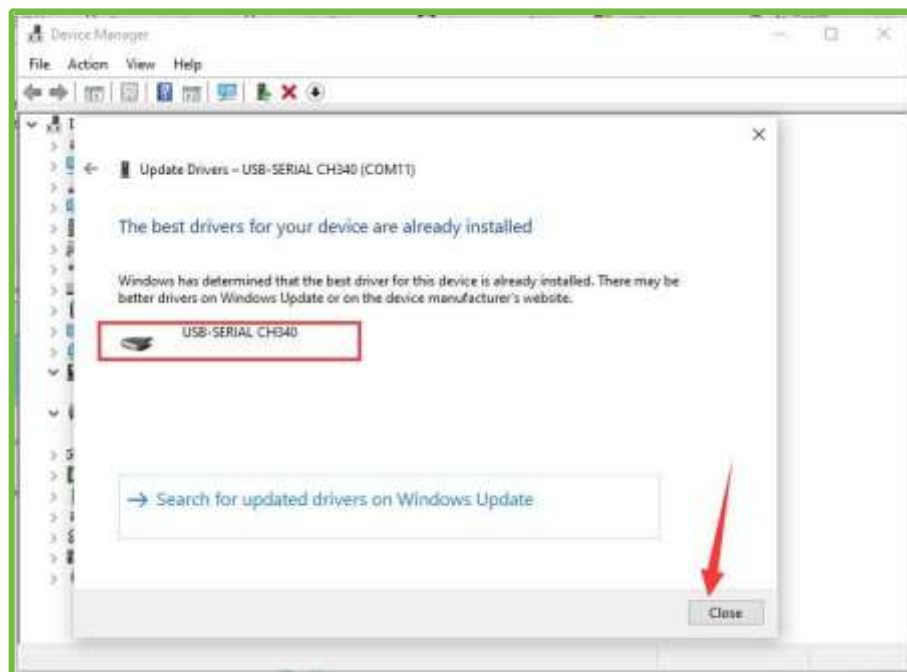


Krok 3: Klikněte na tlačítko "Procházet" a procházejte složky.

vyhledejte umístění souboru ovladače v počítači, klikněte na tlačítko "OK" a nakonec kliknutím na tlačítko "Další" ovladač nainstalujte.

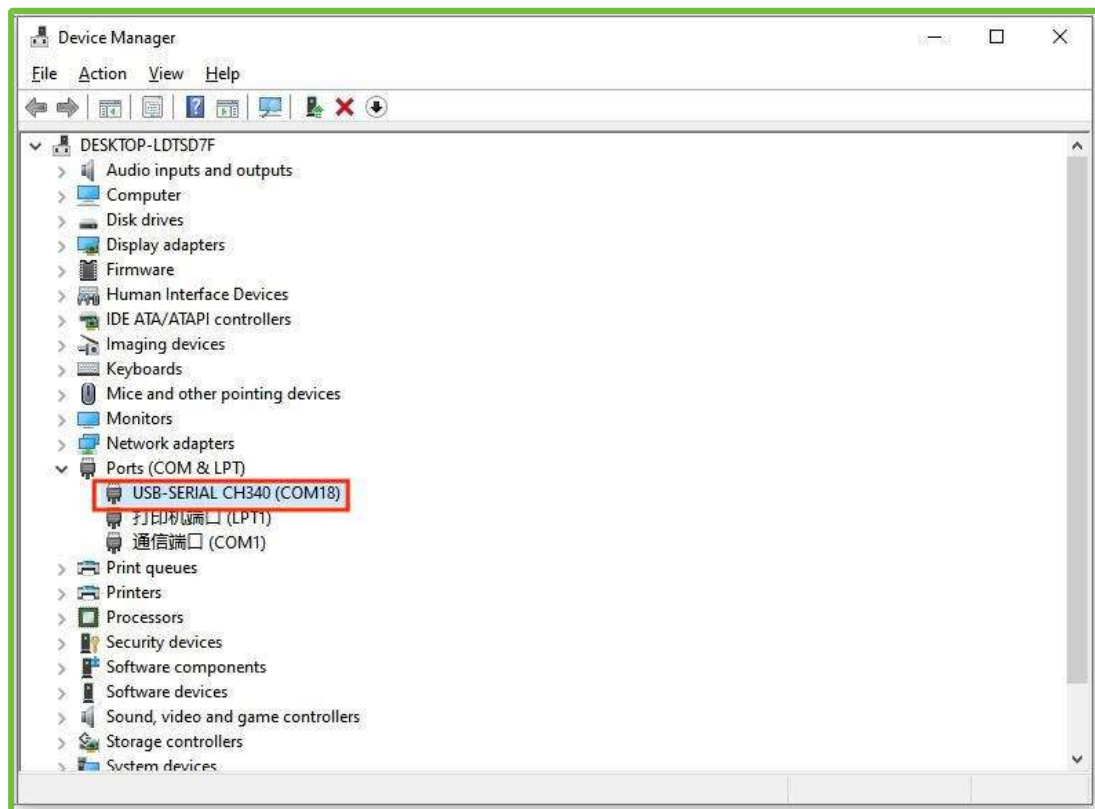


Krok 4: Po dokončení instalace ovladače se zobrazí potvrzovací zpráva. Po dokončení instalace klikněte na tlačítko "Zavřít".



Krok 5: Zkontrolujte, zda je nainstalován sériový ovladač CH340.

úspěšně. Jeden konec kabelu USB zasuňte do řídicí desky ESP32 a druhý konec do portu USB počítače. Klikněte pravým tlačítkem myši na "Můj počítač" -> klikněte na "Vlastnosti" -> klikněte na "Správce zařízení" a po připojení řídicí desky se na následujícím obrázku zobrazí, že instalace proběhla úspěšně.



## Sledujte nás

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

Máme velmi rozsáhlou komunitu, která je velmi nápomocná při řešení problémů, a máme také připravený tým podpory, který vám zodpoví všechny dotazy.



FB skupina ACEBOTT QR kód



QR kód YouTube



0755 23956813

:11 ACEBOTT STEM Education Tech Co.,Ltd.

306, budova 212, Tairan Science Park, Terra 4th Rd, Futian District, ShenZhen, Cina, 518042