



**ZÁKLADNÍ NÁVOD PRO
STARTOVACÍ SADU
MEGA 2560**

Lekce 0 Instalace IDE

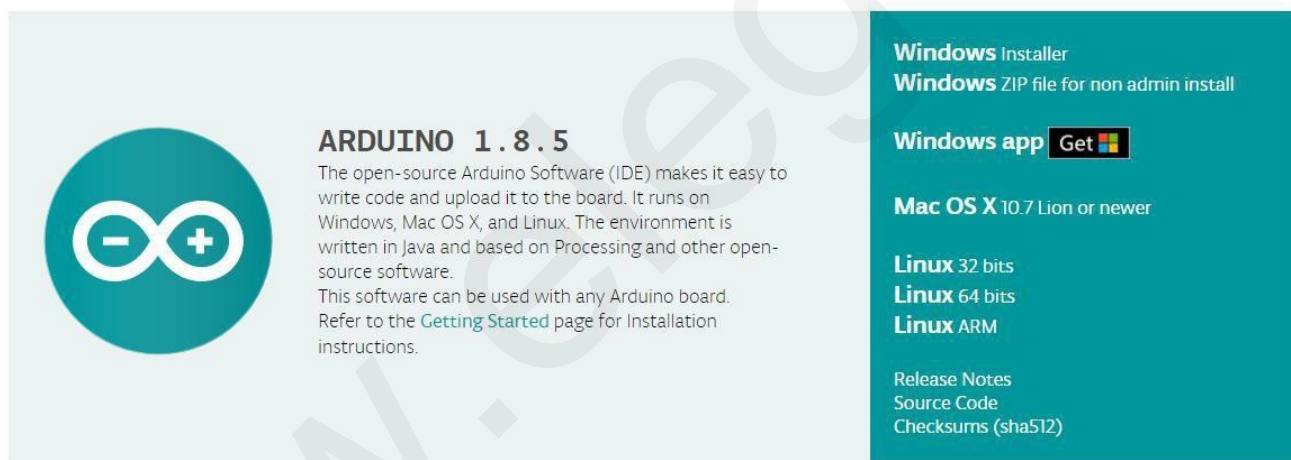
Úvod

Integrované vývojové prostředí (IDE) Arduino je softwarová část platformy Arduino.

V této lekci se dozvíte, jak nastavit počítač pro použití Arduina a jak se pustit do následujících lekcí.

Software Arduino, který budete používat k programování Arduina, je k dispozici pro Windows, Mac a Linux. Instalační proces je pro všechny tři platformy odlišný a bohužel je nutné software nainstalovat ručně.

KROK 1: Přejděte na stránku <https://www.arduino.cc/en/Main/Software> a najděte níže uvedenou stránku.



ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app [Get](#)

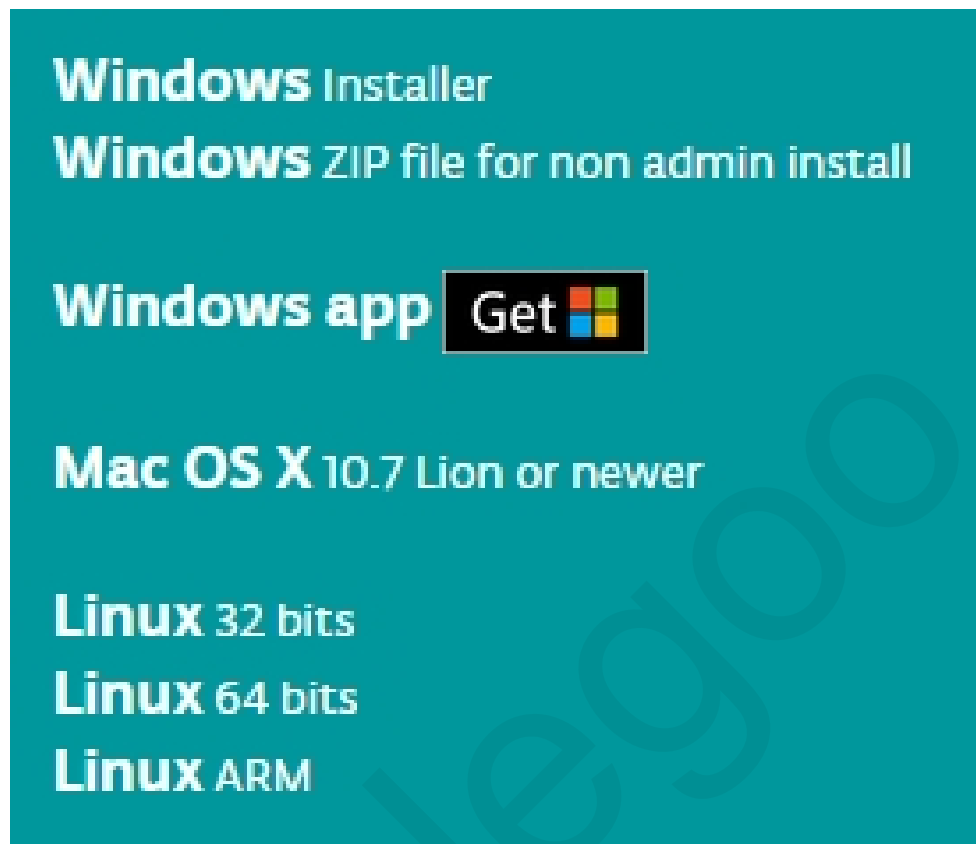
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Verze dostupná na těchto webových stránkách je obvykle nejnovější verze a skutečná verze může být novější než verze na obrázku.

KROK2: Stáhněte si vývojový software, který je kompatibilní s operačním systémem systému počítače. Vezměme si jako příklad systém Windows.



Klikněte na *Instalační program systému Windows*.

Support the Arduino Software

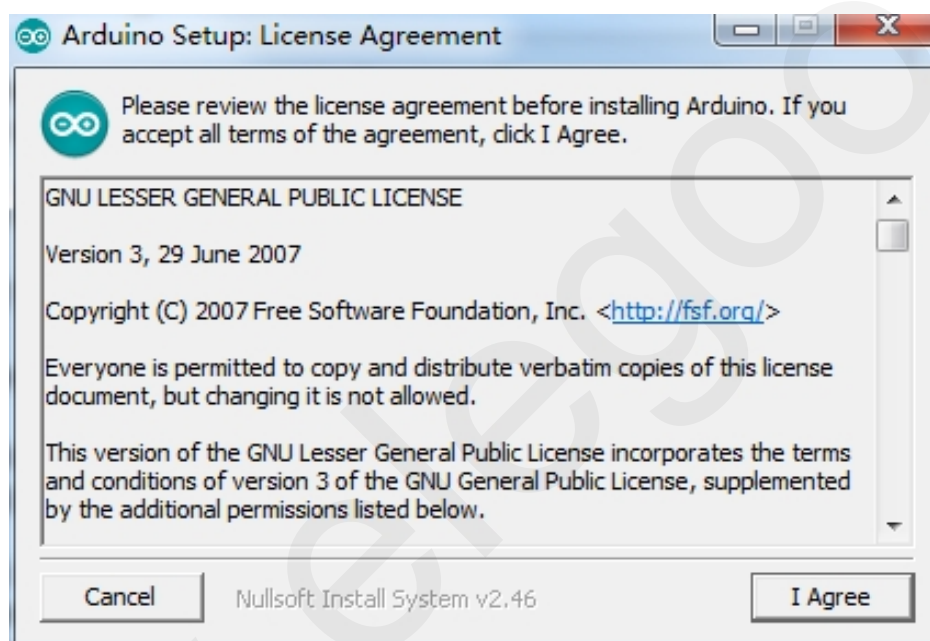
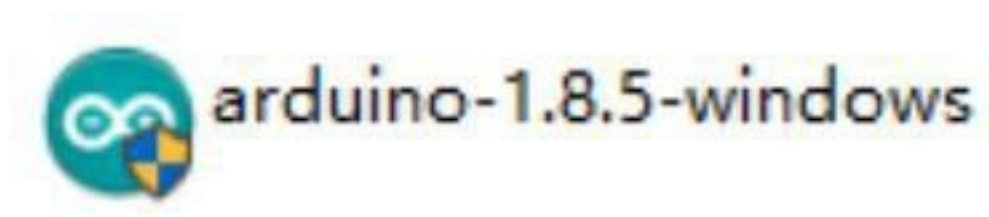
Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



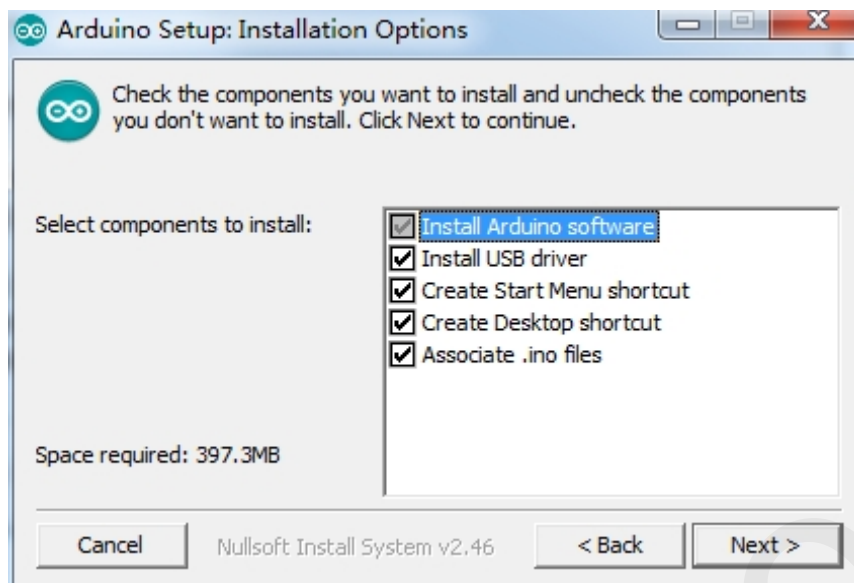
Klikněte na *JUST DOWNLOAD*.

Instalace Arduina (Windows)

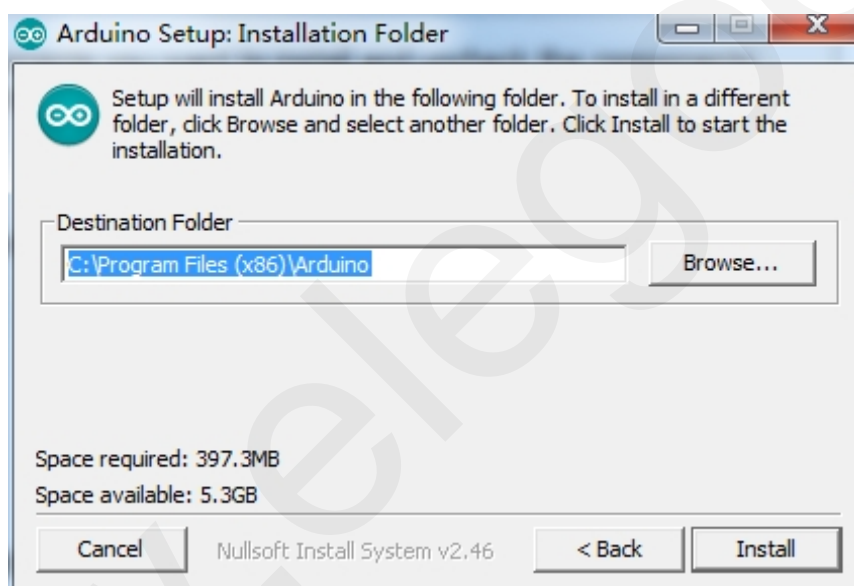
Nainstalujte Arduino pomocí exe. Instalační balíček.



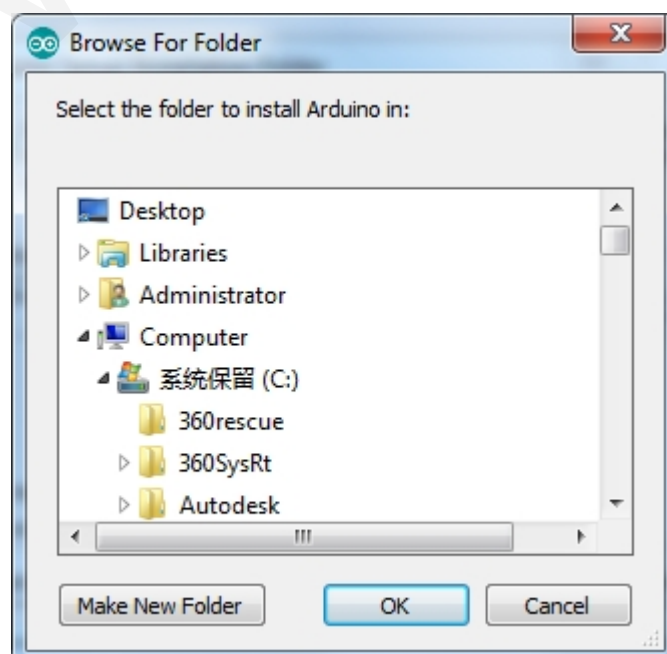
Kliknutím na tlačítko *Souhlasím se* zobrazí následující rozhraní



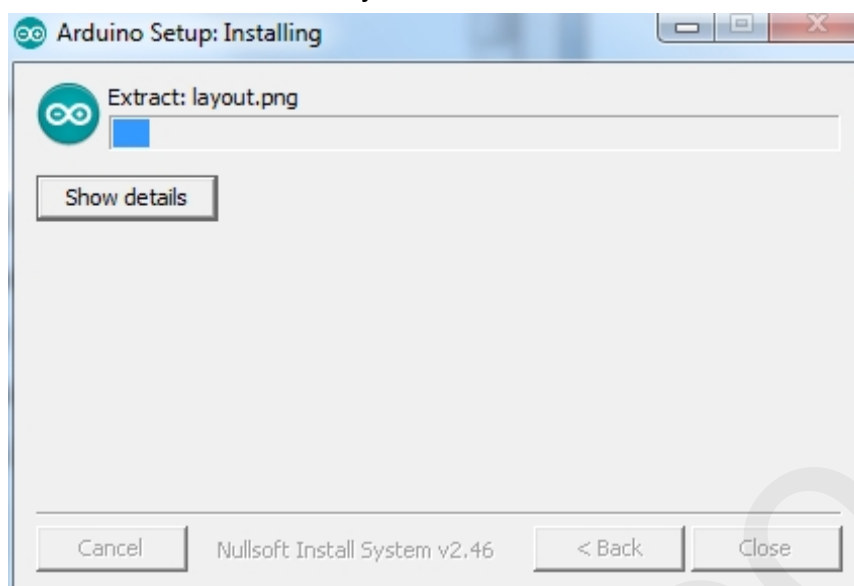
Klikněte na tlačítko *Další*



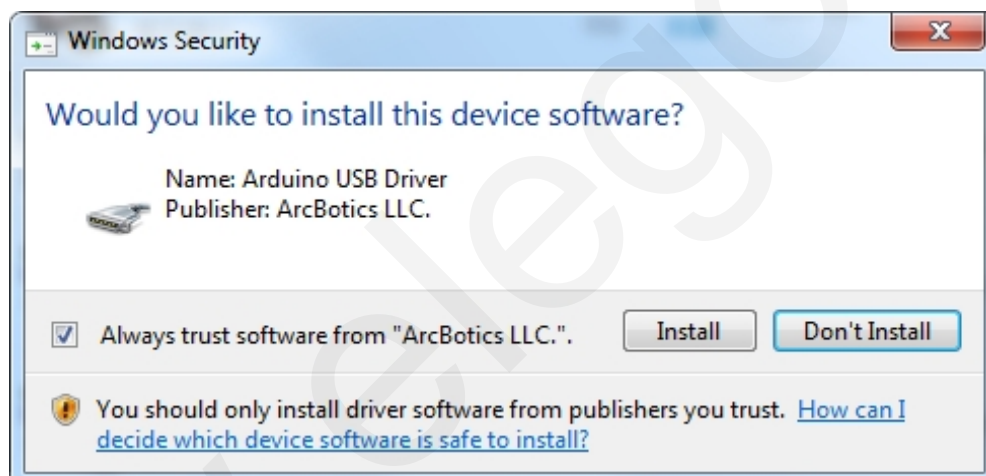
Stisknutím tlačítka **Procházet...** můžete zvolit cestu k instalaci nebo přímo zadat požadovaný adresář.



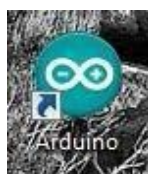
Kliknutím na tlačítko *Instalovat* zahájíte instalaci



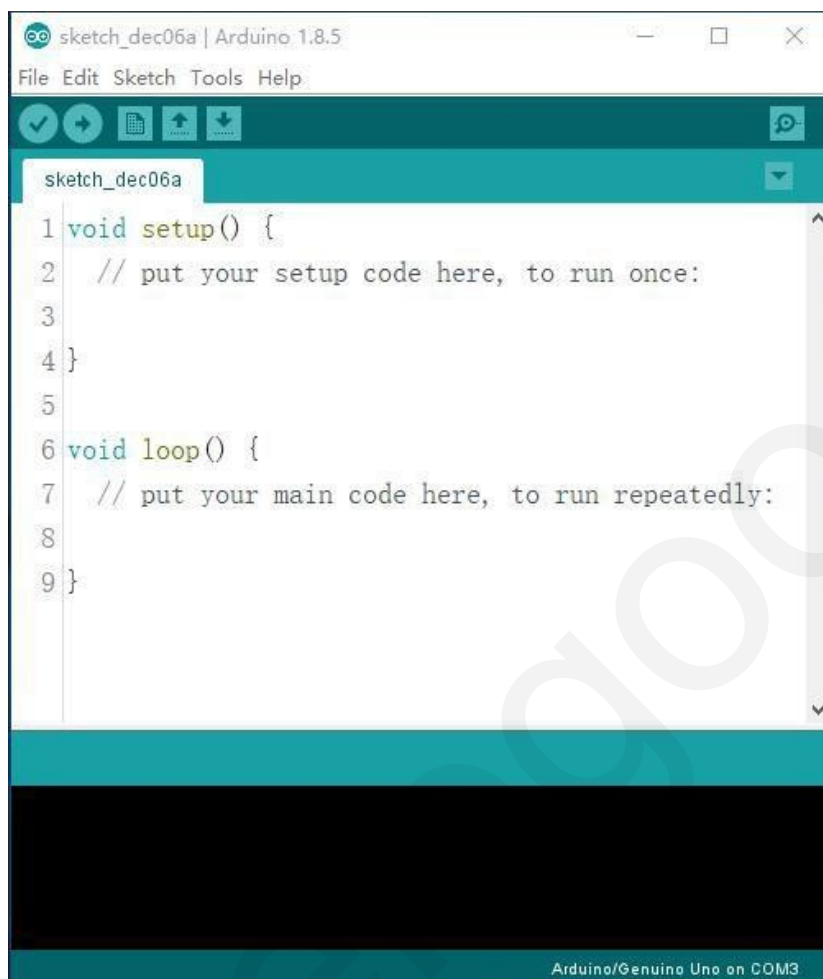
Nakonec se zobrazí následující rozhraní, instalaci dokončete kliknutím na tlačítko *Instalovat*.



Poté se na ploše zobrazí následující ikona

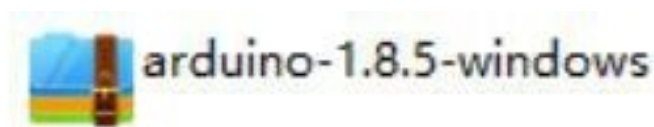


Dvojklikem vstoupíte do požadovaného vývojového prostředí



Můžete přímo vybrat instalační balíček pro instalaci a přeskočit obsah níže a přejít na další část. Pokud se však chcete seznámit s jinými metodami než s instalačním balíčkem, pokračujte ve čtení této části.

Rozbalte stažený soubor zip, dvojitým kliknutím otevřete program a zadejte požadované vývojové prostředí.



此电脑 > Storage (D:) > arduino 1.8.5 > arduino-1.8.5-windows > arduino-1.8.5

名称	修改日期	类型	大小
drivers	2017/12/6 10:15	文件夹	
examples	2017/12/6 10:15	文件夹	
hardware	2017/12/6 10:15	文件夹	
java	2017/12/6 10:15	文件夹	
lib	2017/12/6 10:15	文件夹	
libraries	2017/12/6 10:15	文件夹	
reference	2017/12/6 10:15	文件夹	
tools	2017/12/6 10:15	文件夹	
tools-builder	2017/12/6 10:15	文件夹	
arduino	2017/10/2 15:37	应用程序	395 KB
arduino.l4j	2017/10/2 15:37	配置设置	1 KB
arduino_debug	2017/10/2 15:37	应用程序	393 KB
arduino_debug.l4j	2017/10/2 15:37	配置设置	1 KB
arduino-builder	2017/10/2 15:37	应用程序	3,214 KB
libusb0.dll	2017/10/2 15:37	应用程序扩展	43 KB
msvcp100.dll	2017/10/2 15:37	应用程序扩展	412 KB
msvcr100.dll	2017/10/2 15:37	应用程序扩展	753 KB
revisions	2017/10/2 15:37	文本文档	84 KB
wrapper-manifest	2017/10/2 15:37	XML 文档	1 KB

sketch_dec06a | Arduino 1.8.5

File Edit Sketch Tools Help

sketch_dec06a

```

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }

```

Arduino/Genuino Uno on COM3

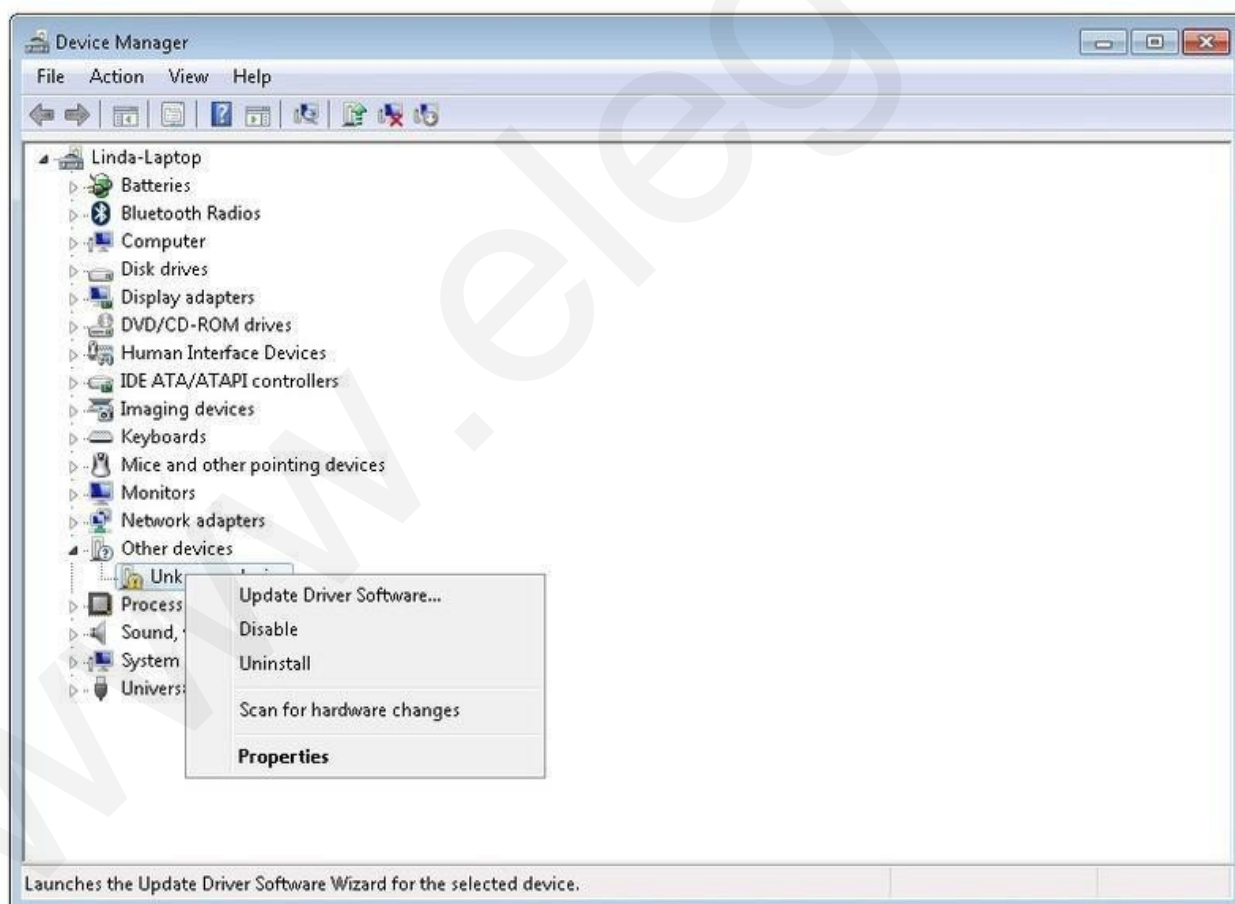
Tento způsob instalace však vyžaduje samostatnou instalaci ovladače.

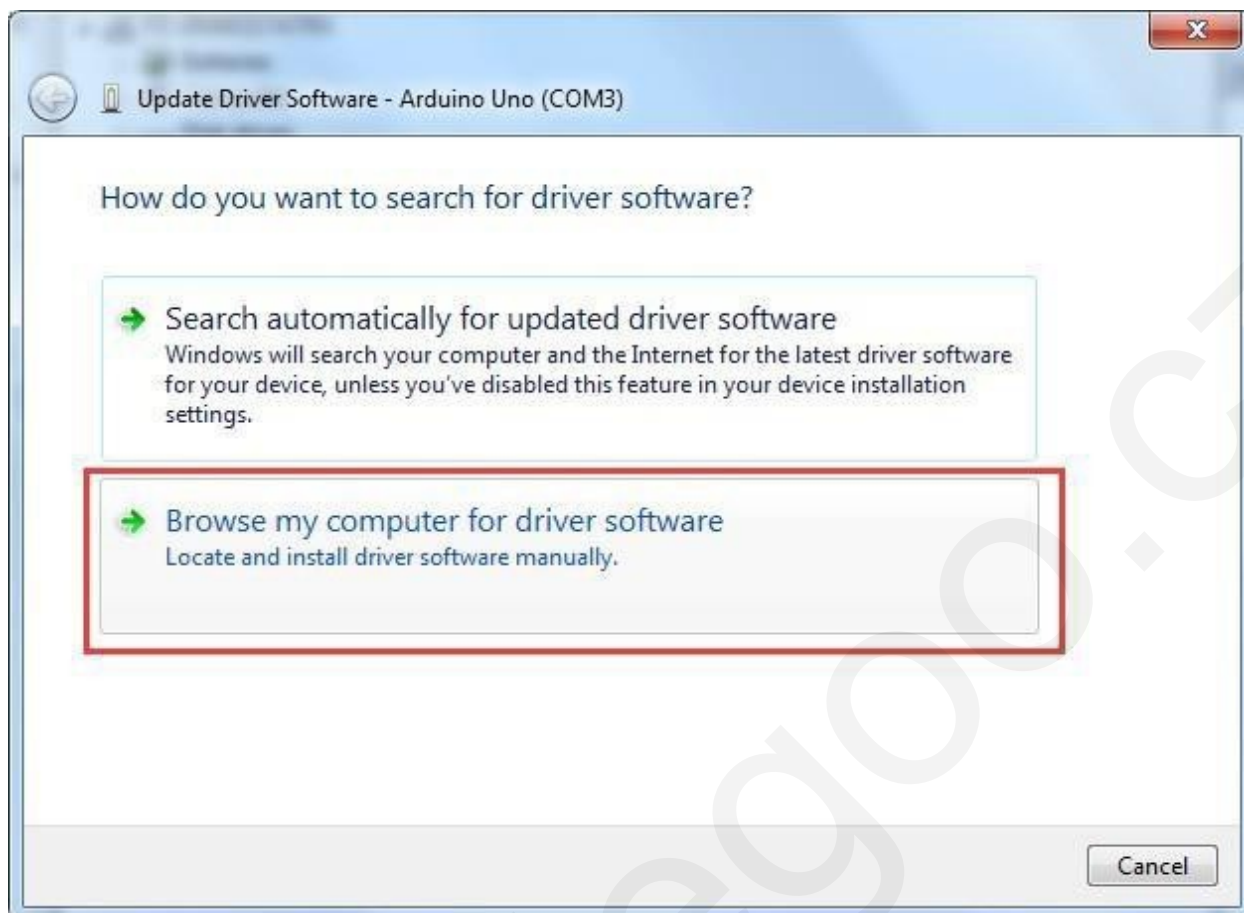
Složka Arduino obsahuje jak samotný program Arduino, tak ovladače, které umožňují připojení Arduina k počítači pomocí kabelu USB. Než spustíme program Arduino, nainstalujete ovladače USB.

Jeden konec kabelu USB zapojte do Arduina a druhý do zásuvky USB v počítači. Na LED diodě se rozsvítí kontrolka napájení a systém Windows může zobrazit zprávu "Found New Hardware" (Nalezen nový hardware). Tuto zprávu ignorujte a zrušte všechny pokusy systému Windows o automatickou instalaci ovladačů.

Nejspolehlivější metodou instalace ovladačů USB je použití Správce zařízení. Ten je přístupný různými způsoby v závislosti na verzi systému Windows. V systému Windows 7 musíte nejprve otevřít Ovládací panely, poté vybrat možnost zobrazení ikon a v seznamu byste měli najít Správce zařízení.

V části "Ostatní zařízení" byste měli vidět ikonu "neznámého zařízení" s malým žlutým výstražným trojúhelníčkem vedle. To je vaše Arduino.





Klikněte pravým tlačítkem myši na zařízení a vyberte možnost v horní nabídce (Aktualizovat software ovladače...). Poté budete vyzváni k volbě "Automaticky vyhledat aktualizovaný software ovladače" nebo "Procházet můj počítač pro software ovladače". Vyberte možnost procházet a přejděte na X\arduino1.8.0\drivers.



Klikněte na tlačítko "Další" a možná se zobrazí bezpečnostní varování, pokud ano, povolte instalaci softwaru. Po instalaci softwaru se zobrazí potvrzovací zpráva.



Uživatelé systému Windows mohou přeskočit pokyny k instalaci pro systémy Mac a Linux a přejít na lekci 1. Uživatelé systémů Mac a Linux mohou pokračovat ve čtení této části.

Instalace Arduina (Mac OS X)

První krok: Stáhněte si software Arduino (IDE)

-Otevřít adresu URL: <https://www.arduino.cc/en/Main/Software> pomocí prohlížeče



ARDUINO 1.8.5
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows installer
Windows ZIP file for non admin install

Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

-klikněte na Mac OSX 10.7 Lion nebo novější

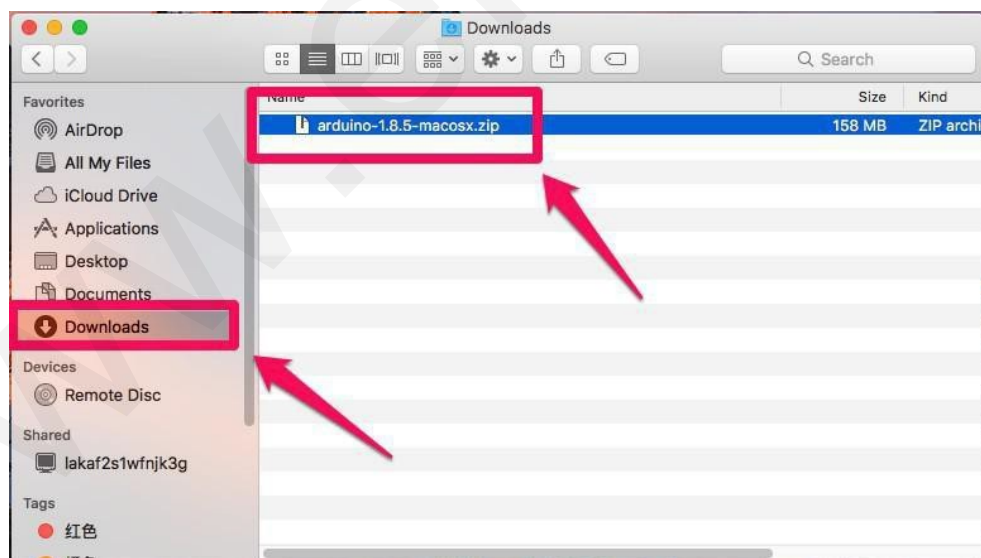
Support the Arduino Software

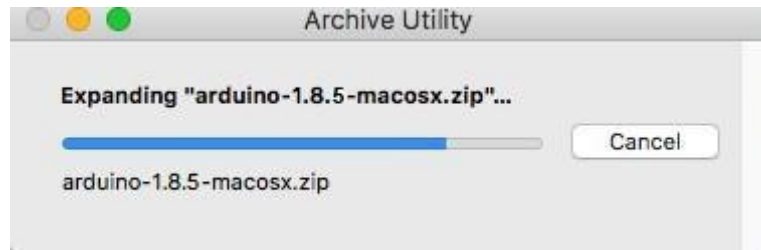
Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



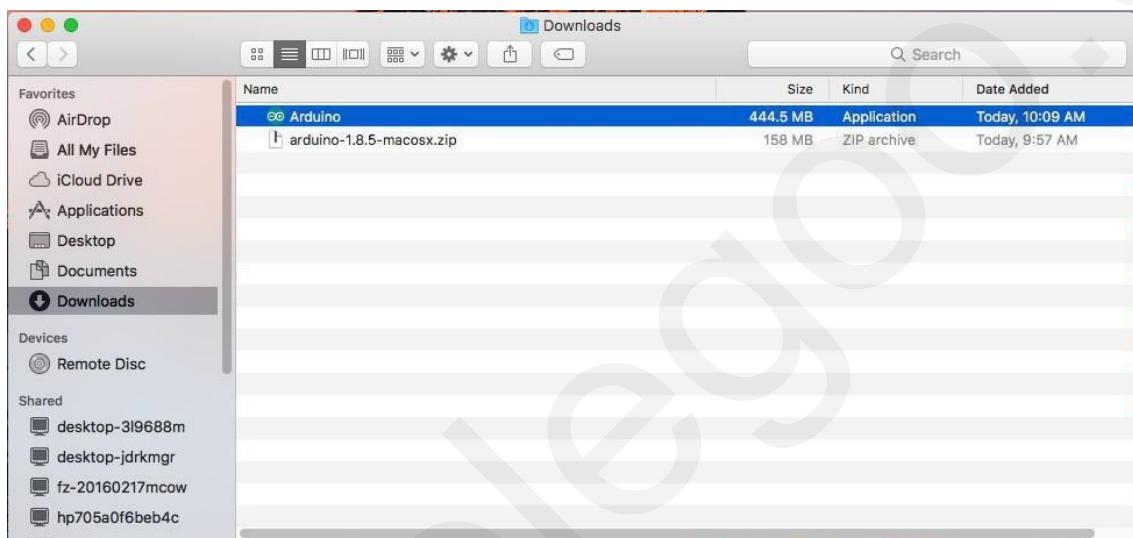
-klikněte na možnost JEN STÁHNOUT

Stáhněte a rozbalte soubor zip, dvakrát klikněte na Arduino. app a vstupte do prostředí Arduino IDE; systém vás požádá o instalaci knihovny Java runtime, pokud ji v počítači nemáte. Po dokončení instalace můžete spustit Arduino IDE.

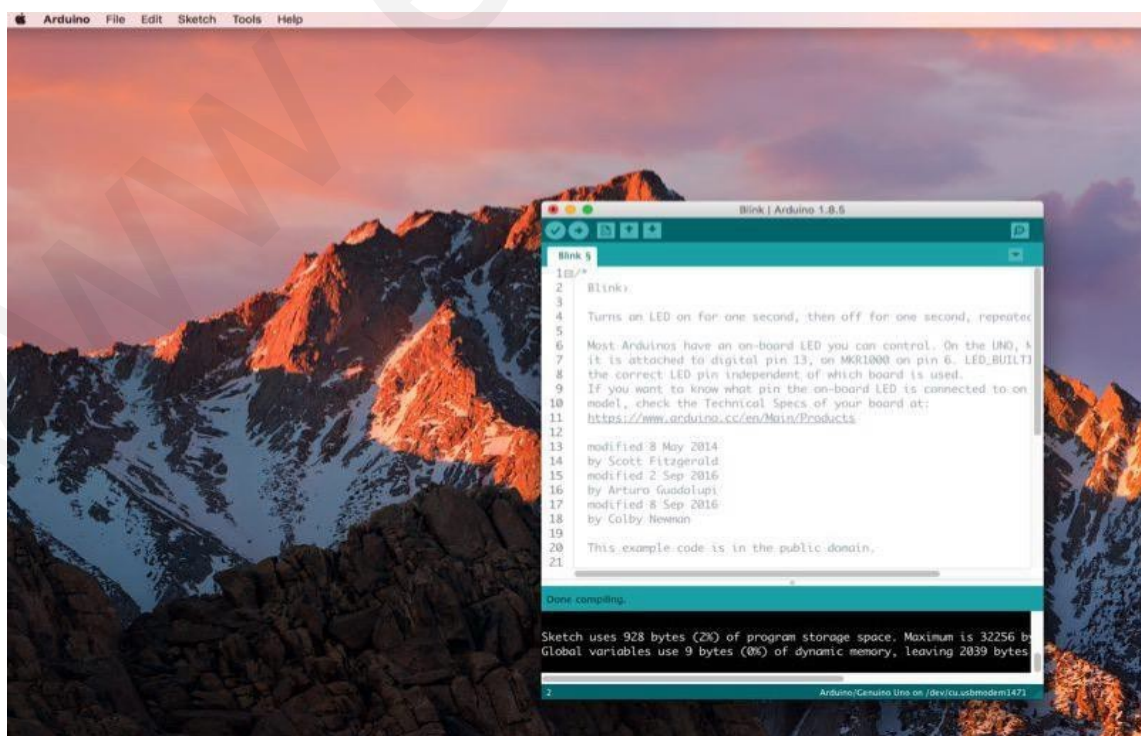




-arduino soubor je aplikace Arduino, stačí na ni dvakrát kliknout a otevře se.



-Arduino IDE



Instalace Arduina (Linux)

Musíte použít příkaz "make install". Pokud používáte systém Ubuntu, doporučujeme nainstalovat Arduino IDE z Centra softwaru Ubuntu.

Můžete se podívat na následující odkazy, které nabízejí konkrétní kroky instalace.

<https://www.arduino.cc/en/guide/linux>



INSTALL ARDUINO ON LINUX.pdf



arduino-1.8.5-linux32.tar.xz



arduino-1.8.5-linux64.tar.xz

Lekce 1 Přidání knihoven a otevření SerialMonitoru

Instalace dalších knihoven ArduinoLibraries

Jakmile se seznámíte se softwarem Arduino a používáte vestavěné funkce, můžete chtít rozšířit možnosti svého Arduina o další knihovny.

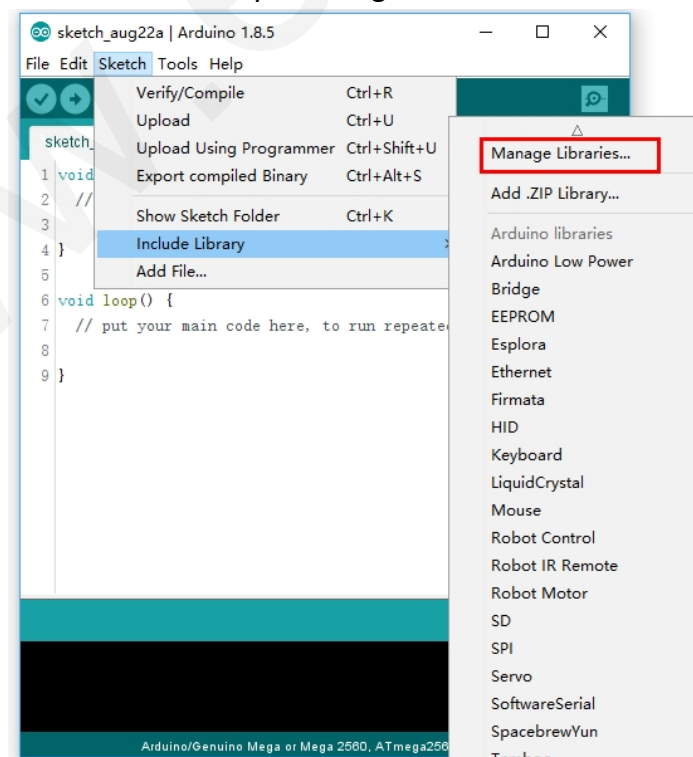
Co jsou knihovny?

Knihovny představují soubor kódu, který usnadňuje připojení k senzoru, displeji, modulu atd. Například vestavěná knihovna Liquid Crystal usnadňuje komunikaci se znakovými LCD displeji. Na internetu jsou ke stažení stovky dalších knihoven. Seznam vestavěných knihoven a některých těchto doplňkových knihoven je uveden v referenci. Chcete-li doplňkové knihovny používat, musíte si je nainstalovat.

Jak nainstalovat knihovnu

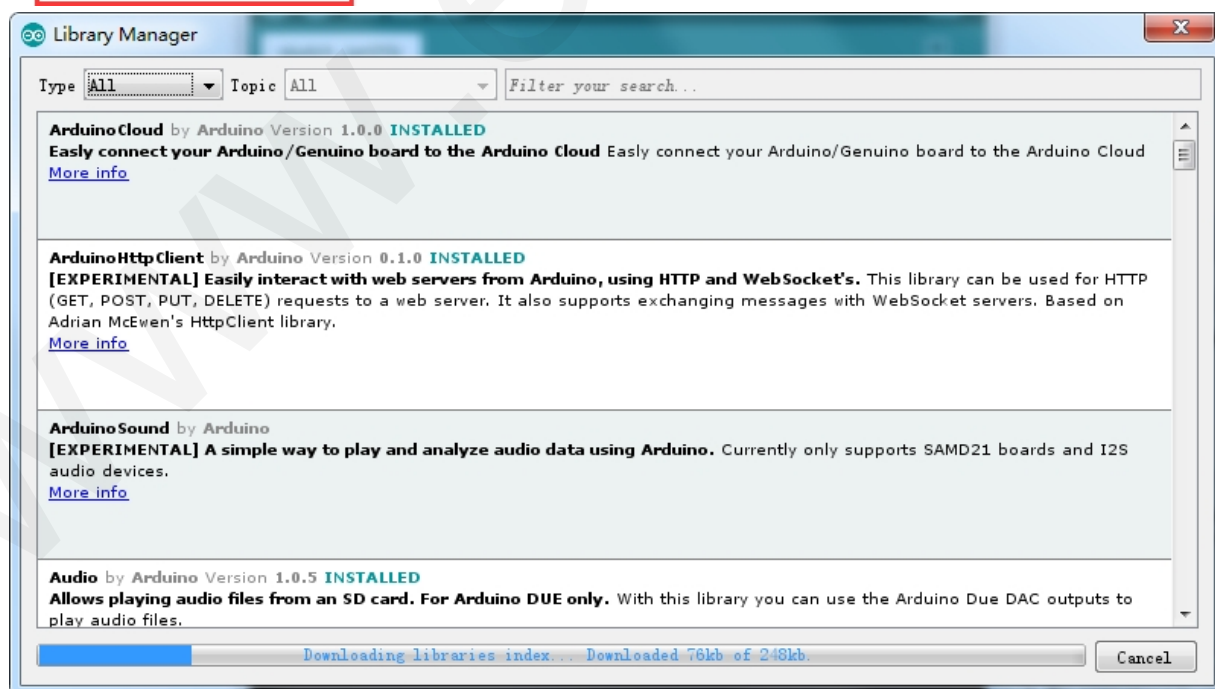
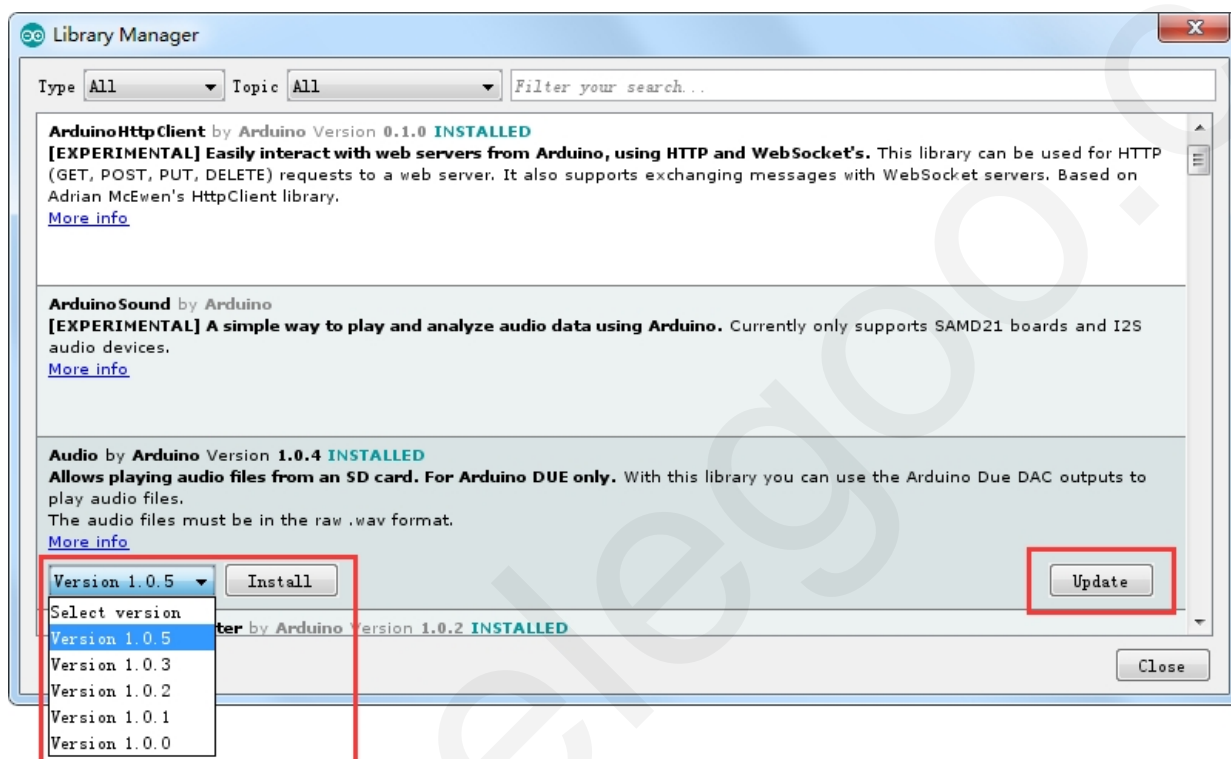
Používání Správce knihovny

Pro instalaci nové knihovny do prostředí Arduino IDE můžete použít Správce knihoven (dostupný od verze IDE 1.8.0). Otevřete IDE a klikněte na nabídku "Sketch" a poté na Include Library > Manage Libraries.

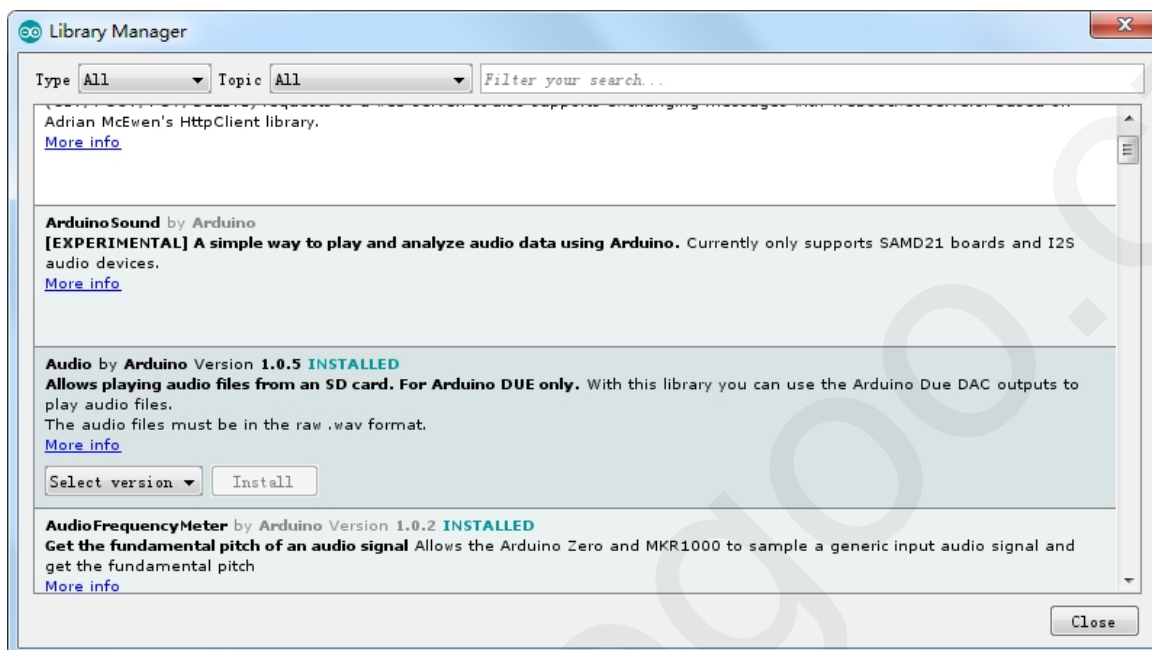


Poté se otevře správce knihoven a zobrazí se seznam knihoven, které jsou již nainstalovány nebo připraveny k instalaci. V tomto příkladu nainstalujeme knihovnu Bridge. Projděte seznam, abyste ji našli, a poté vyberte verzi knihovny, kterou chcete nainstalovat. Někdy je k dispozici pouze jedna verze knihovny. Pokud se nabídka pro výběr verze nezobrazí, neznepokojujte se: je to normální.

Někdy s ním musíte být trpěliví, jak je znázorněno na obrázku. Obnovte jej a vyčkejte.



Nakonec klikněte na tlačítko install a počkejte, až IDE nainstaluje novou knihovnu. Stahování může trvat delší dobu v závislosti na rychlosti vašeho připojení. Po dokončení by se vedle knihovny Bridge měla objevit značka Installed. Správce knihoven můžete zavřít.



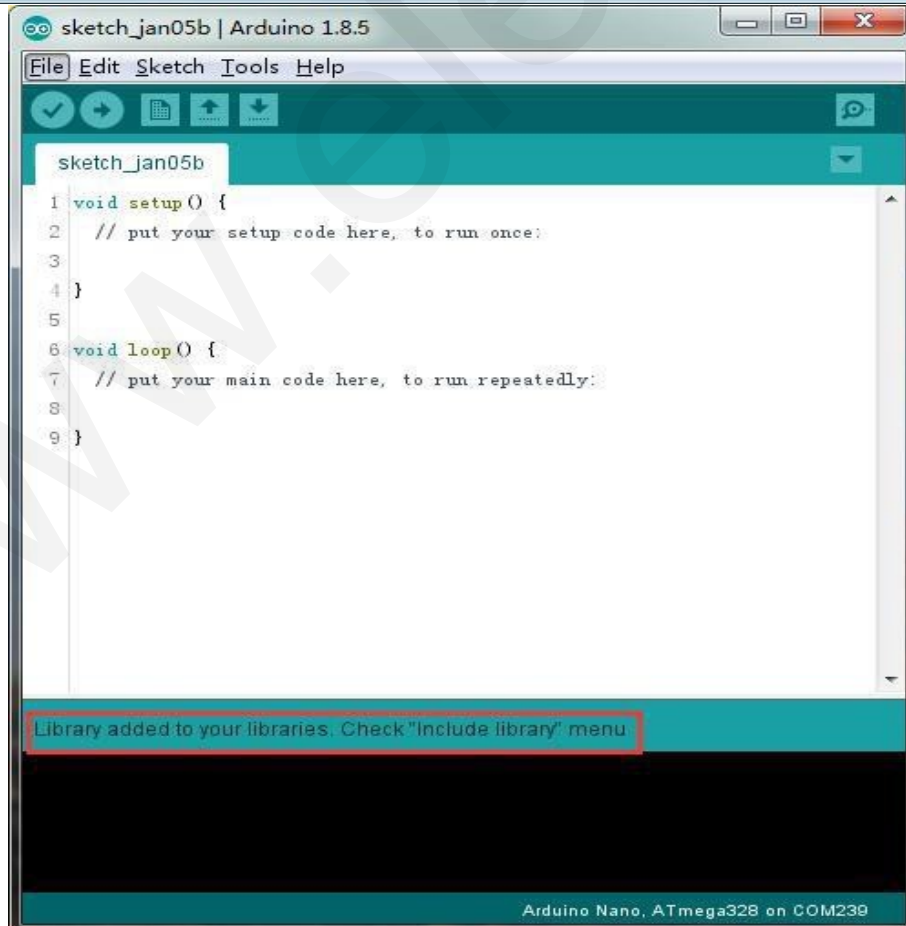
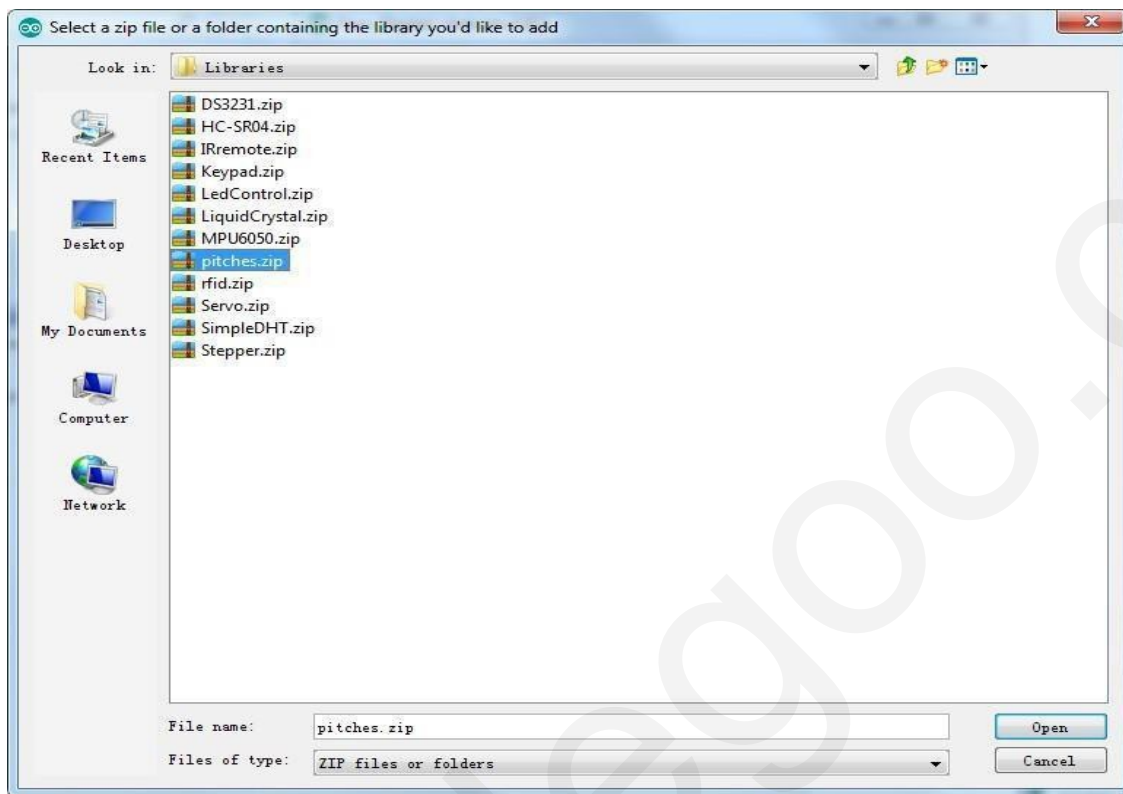
Novou knihovnu nyní najdete v nabídce Zahrnout knihovnu. Pokud chcete přidat vlastní knihovnu, otevřete nový problém na [Githubu](#).

Importování knihovny .zip

Knihovny jsou často distribuovány jako soubor ZIP nebo složka. Název složky je název knihovny. Uvnitř složky se nachází soubor .cpp, soubor .h a často také soubor keywords.txt, složka examples a další soubory požadované knihovnou. Počínaje verzí 1.0.5 můžete v prostředí IDE instalovat knihovny třetích stran. Staženou knihovnu nerozbalujte, nechte ji tak, jak je.

V prostředí Arduino IDE přejděte do nabídky Sketch > Include Library. V horní části rozevřacího seznamu vyberte možnost "Přidat knihovnu .ZIP".

Budete vyzváni k výběru knihovny, kterou chcete přidat. Přejděte do umístění souboru .zip a otevřete jej.



Vraťte se do nabídky Sketch > Import Library. Nyní byste měli vidět knihovnu v dolní části rozbalovací nabídky. Je připravena k použití ve vašem náčrtu. Soubor ZIP bude rozbalen ve složce knihoven v adresáři Arduino sketches. **Poznámka:**

knihovna bude k dispozici pro použití ve skicách, ale příklady pro knihovnu budou vystaveny v nabídce Soubor > Příklady až po restartu IDE.

Tyto dva přístupy jsou nejčastější. Podobně lze postupovat i v systémech MAC a Linux. Ruční instalace, která bude představena níže jako alternativa, se může používat jen zřídka a uživatelé bez potřeb ji mohou přeskočit.

Ruční instalace

Chcete-li knihovnu nainstalovat, nejprve ukončete aplikaci Arduino. Poté rozbalte soubor ZIP obsahující knihovnu . Pokud například instalujete knihovnu nazvanou "ArduinoParty", rozbalte soubor ArduinoParty.zip. Měl by obsahovat složku s názvem ArduinoParty a v ní soubory jako ArduinoParty.cpp a ArduinoParty.h. (Pokud soubory .cpp a .h ve složce nejsou, budete ji muset vytvořit. V takovém případě vytvoříte složku s názvem "ArduinoParty" a přesunete do ní všechny soubory, které byly v souboru ZIP, například ArduinoParty.cpp a ArduinoParty.h.)

Do této složky (složky knihoven) přetáhněte složku Arduino Party. Pod Windows se bude pravděpodobně jmenovat "Moje dokumenty\Arduino\libraries". Pro uživatele Macu se bude pravděpodobně jmenovat "Documents/Arduino/libraries". V systému Linux to bude složka "libraries" v aplikaci yoursketchbook.

Vaše složka s knihovnou Arduino by nyní měla vypadat takto (v systému Windows):

MyDocuments\Arduino\libraries\ArduinoParty\ArduinoParty.cpp
My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h
MyDocuments\Arduino\libraries\ArduinoParty\examples

nebo takto (v systémech Mac a Linux):

Dokumenty/Arduino/knihovny/ArduinoParty/ArduinoParty.cpp
Dokumenty/Arduino/knihovny/ArduinoParty/ArduinoParty.h
Dokumenty/Arduino/knihovny/ArduinoParty/příklady

....

www.elegoo.cz

Může existovat více souborů než jen soubory .cpp a .h, jen se ujistěte, že jsou tam všechny. (Knihovna nebude fungovat, pokud soubory .cpp a .h vložíte přímo do složky libraries nebo pokud jsou vnořené v další složce. Např:

Dokumenty\Arduino\libraries\ArduinoParty.cpp

a

Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp

nebudou fungovat.)

Restartujte aplikaci Arduino. Ujistěte se, že se nová knihovna objevila v okně Sketch -> Importovat knihovnu v nabídce softwaru. To je vše! Nainstalovali jste knihovnu!

Arduino Serial Monitor (Windows, Mac, Linux)

Integrované vývojové prostředí (IDE) Arduino je softwarová část platformy Arduino.

A protože používání terminálu je důležitou součástí práce s Arduinem a dalšími mikrokontroléry, rozhodli se do softwaru zahrnout i sériový terminál. V rámci prostředí Arduino se tento terminál nazývá Serial Monitor.

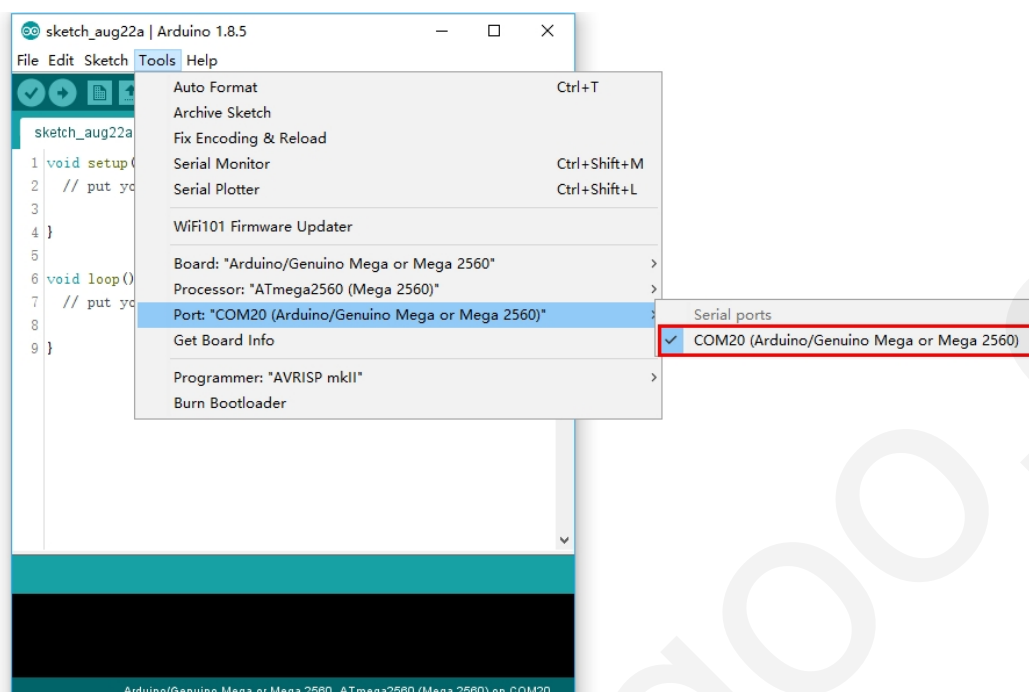
Navázání spojení

Sériový monitor je dodáván se všemi verzemi prostředí Arduino IDE. Chcete-li jej otevřít, stačí kliknout na ikonu Serial Monitor.

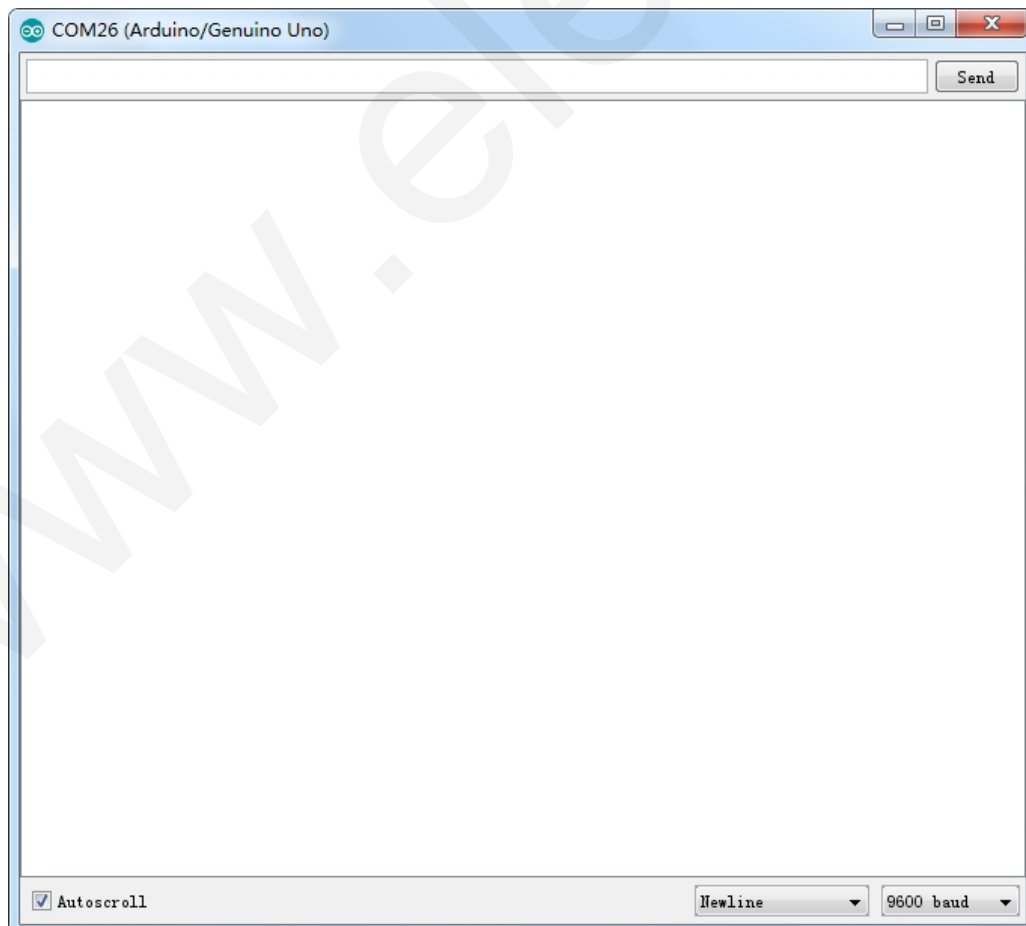


Výběr portu, který se má otevřít v Sériovém monitoru, je stejný jako výběr portu pro nahrávání kódu Arduina. Přejděte na Tools -> Serial Port a vyberte správný port.

Tipy: Zvolte stejný port COM, který máte v DeviceManageru.

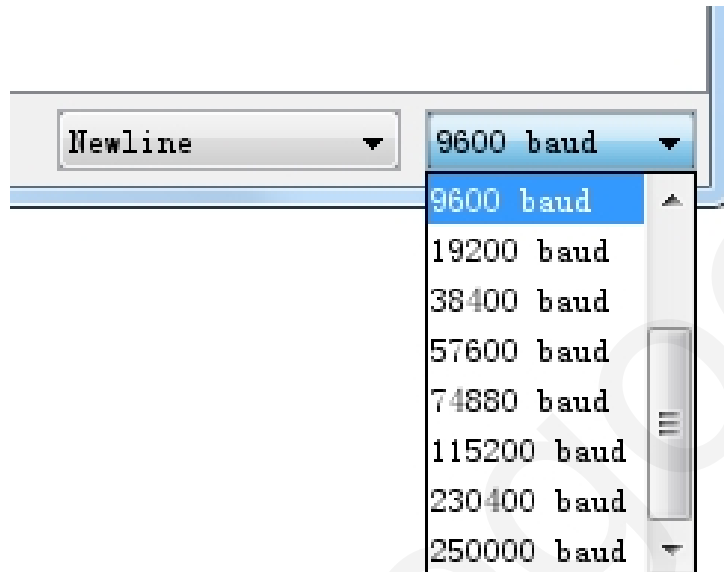


Po otevření byste měli vidět něco podobného:

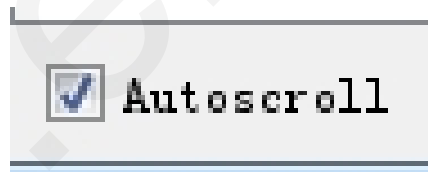


Nastavení

Sériový monitor má omezené možnosti nastavení, ale postačí pro většinu vašich potřeb v oblasti sériové komunikace. První nastavení, které můžete změnit, je přenosová rychlost. Klepnutím na rozevírací nabídku baud rate (přenosová rychlost) vyberte správnou přenosovou rychlost. (9600 baud)



Nakonec můžete zaškrtnutím políčka v levém dolním rohu nastavit automatické posouvání terminálu.



Klady

Sériový monitor je skvělý rychlý a snadný způsob, jak navázat sériové spojení s vaším Arduinem. Pokud již pracujete v prostředí Arduino IDE, není opravdu nutné otevírat samostatný terminál pro zobrazení dat.

Nevýhody

Nedostatek nastavení ponechává Serial Monitor bez povšimnutí a pro pokročilou sériovou komunikaci nemusí být dostačující.

Lekce 2 Mrknutí

Přehled

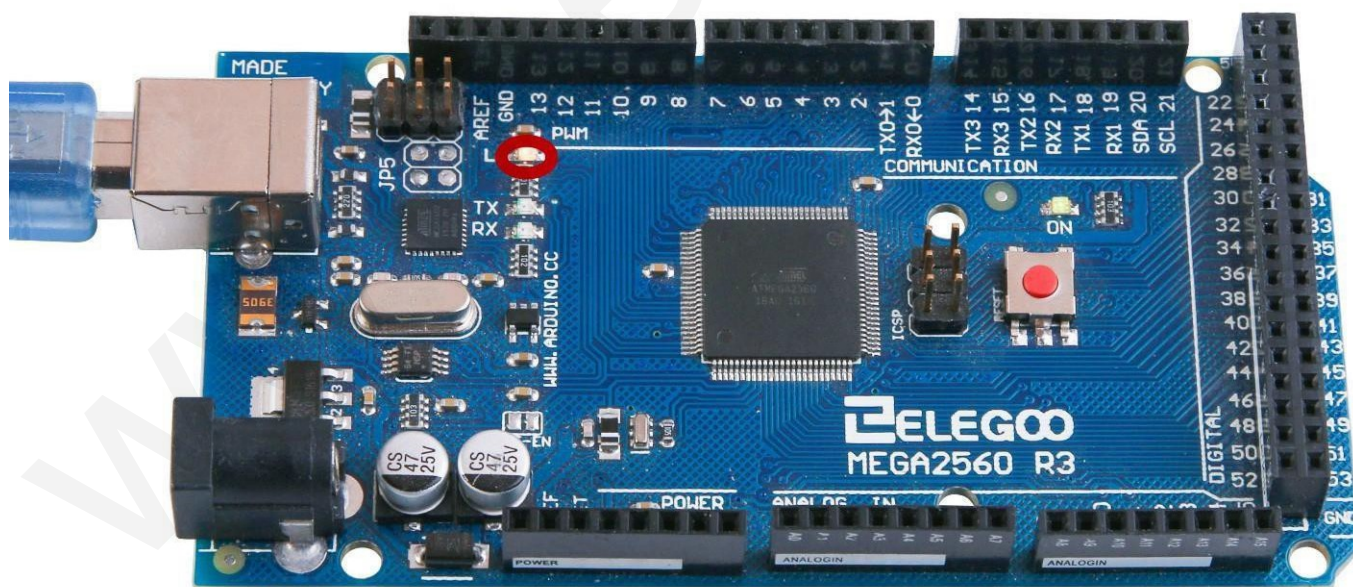
V této lekci se dozvíte, jak naprogramovat řídicí desku MEGA 2560 R3 tak, aby blikala vestavěná LED dioda Arduino, a jak v základních krocích stahovat programy.

Požadovaná součást:

(1) x Elegoo Mega 2560R3

Princip

Deska MEGA2560 R3 má po obou stranách řadu konektorů, které slouží k připojení několika elektronických zařízení a zásuvných "štitů", které rozšiřují její možnosti. Má také jednu LED diodu, kterou můžete ovládat ze svých náčrtů. Tato LED dioda je zabudována na desce MEGA2560 R3 a často se označuje jako LED dioda "L", protože je tak na desce označena.



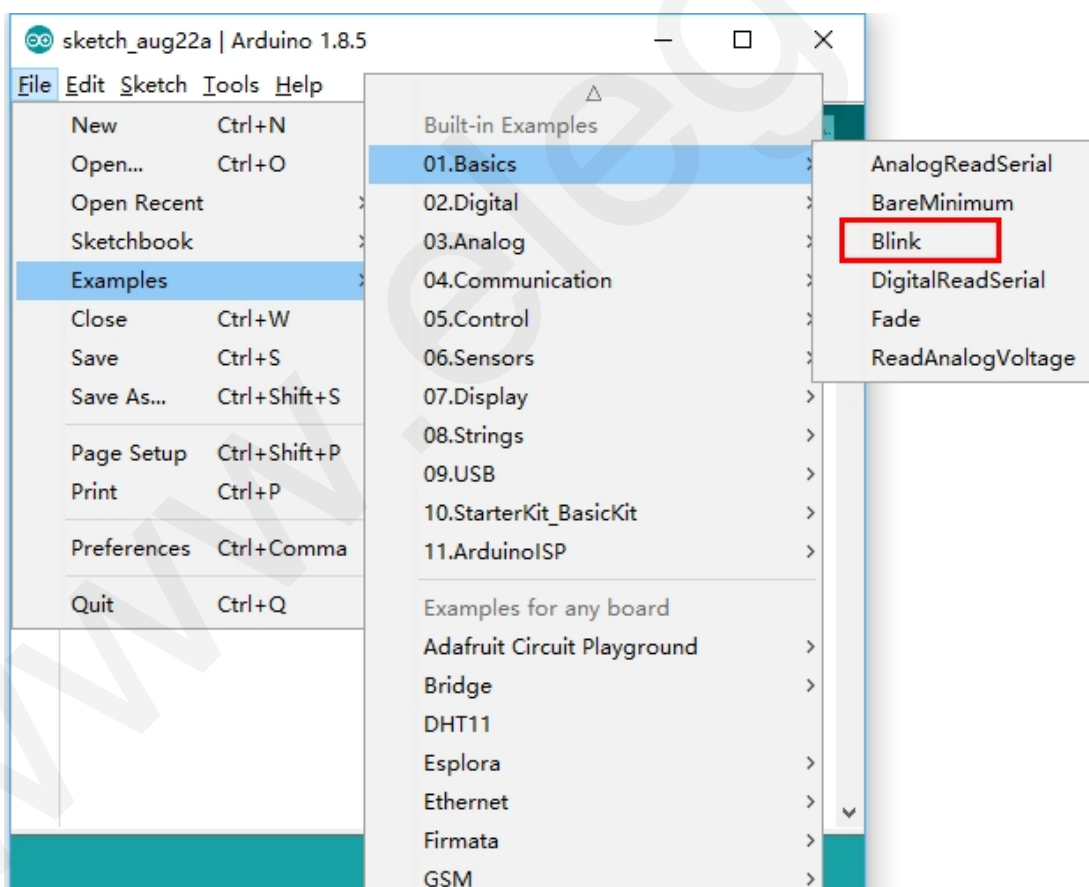
Možná zjistíte, že LED dioda "L" na desce MEGA 2560 R3 bliká již při připojení do zásuvky USB. Je to proto, že desky jsou obvykle dodávány s předinstalovaným skečem 'Blink'.

V této lekci přeprogramujeme desku MEGA 2560 R3 pomocí vlastního skeče Blink a poté změníme rychlost blikání.

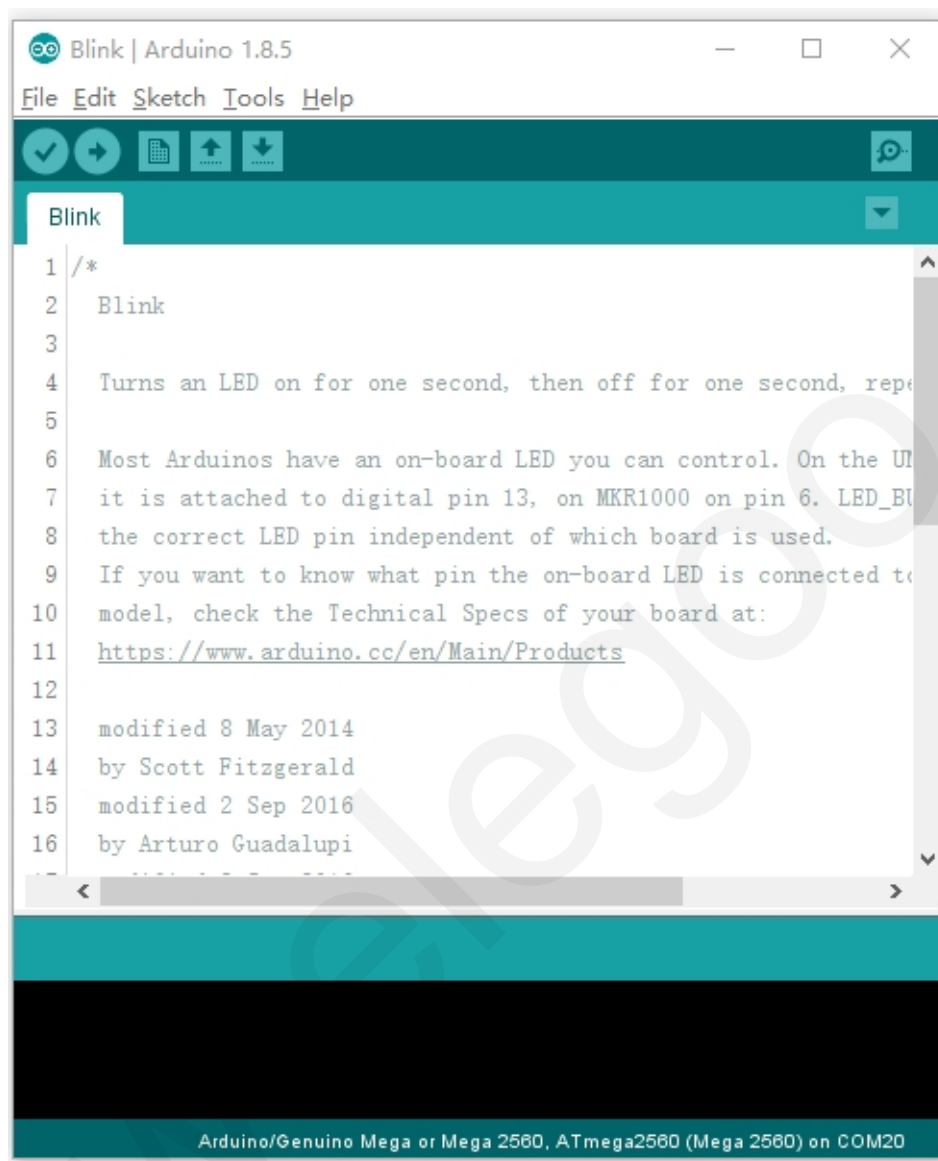
V lekci 0 jste si nastavili Arduino IDE a ujistili se, že jste našli správný sériový port pro připojení k desce MEGA 2560 R3. Nyní nastal čas toto připojení vyzkoušet a naprogramovat desku MEGA 2560 R3.

Arduino IDE obsahuje rozsáhlou sbírku příkladů skic, které můžete načíst a použít. Patří sem i ukázkový skeč pro blikání LED diody "L".

Načtěte skicu "Blink", kterou najdete v nabídce IDE pod položkou File > Examples > 01. Základy



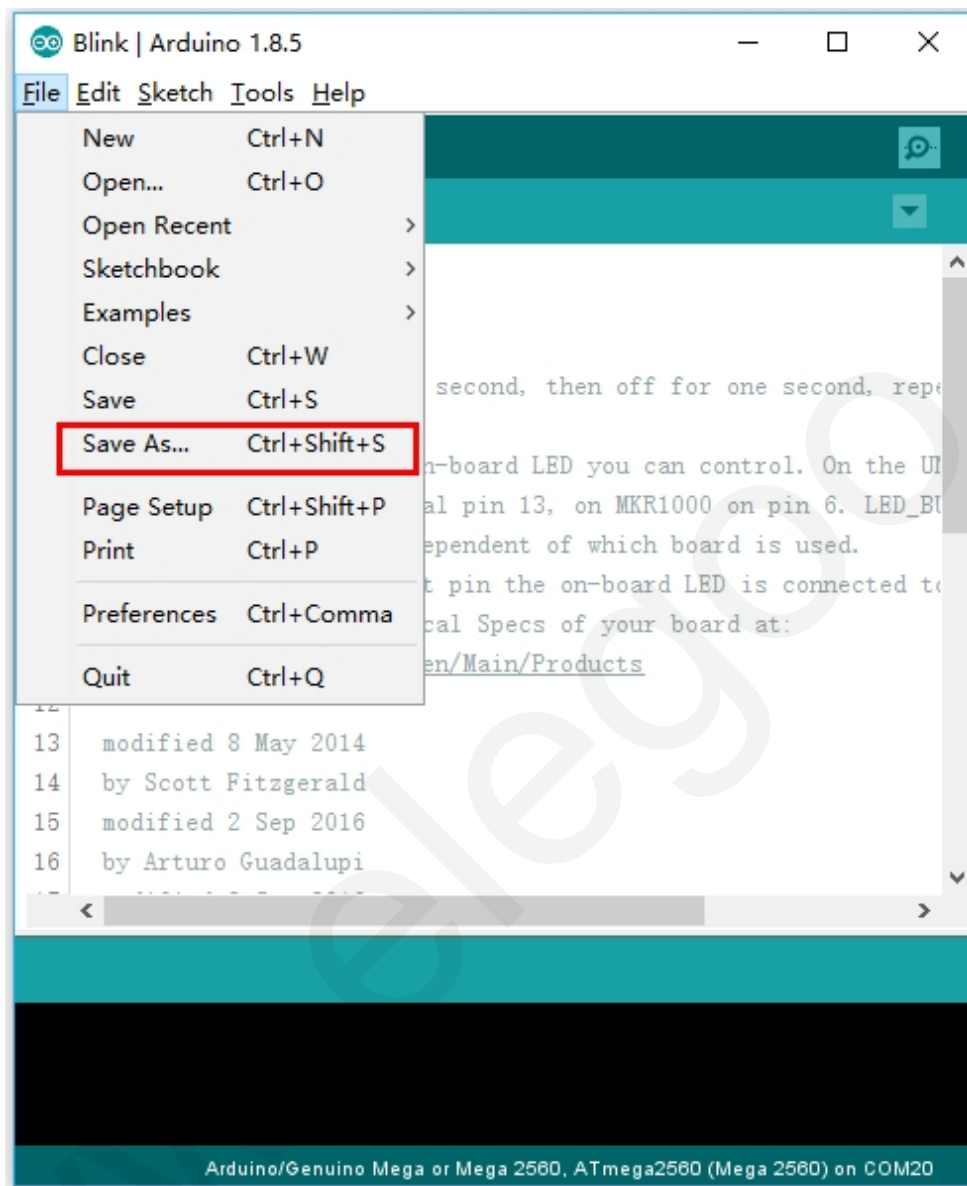
Po otevření okna náčrtu jej zvětšete tak, abyste v něm viděli celý náčrt.

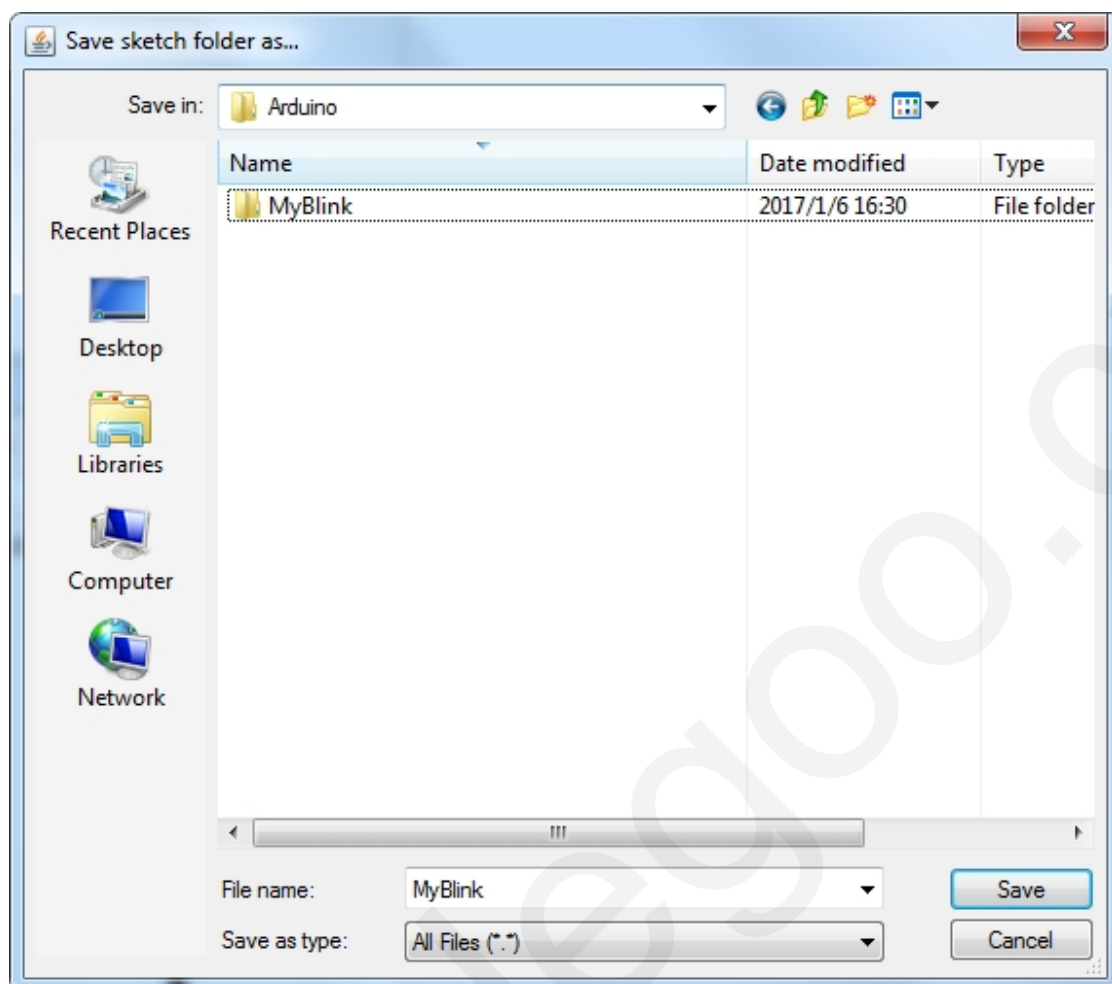


Příkladové náčrty, které jsou součástí Arduino IDE, jsou určeny pouze pro čtení. To znamená, že je můžete nahrát na desku MEGA2560 R3, ale pokud je změníte, nemůžete je uložit jako stejný soubor.

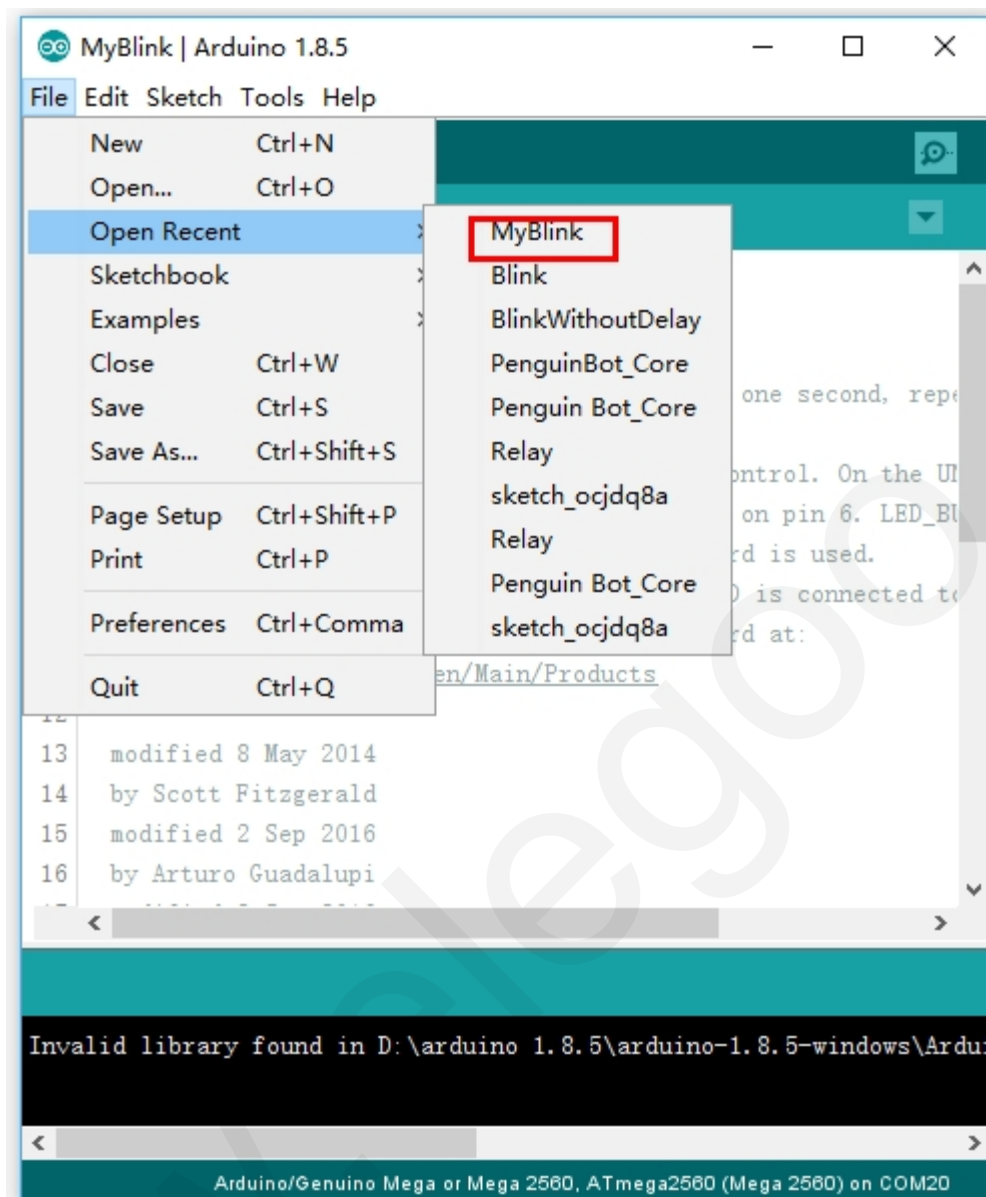
Protože se chystáme tento náčrt změnit, musíte nejprve uložit svou vlastní kopii.

V nabídce Soubor v prostředí Arduino IDE vyberte možnost "Uložit jako.." a poté uložte náčrt s názvem "MyBlink".

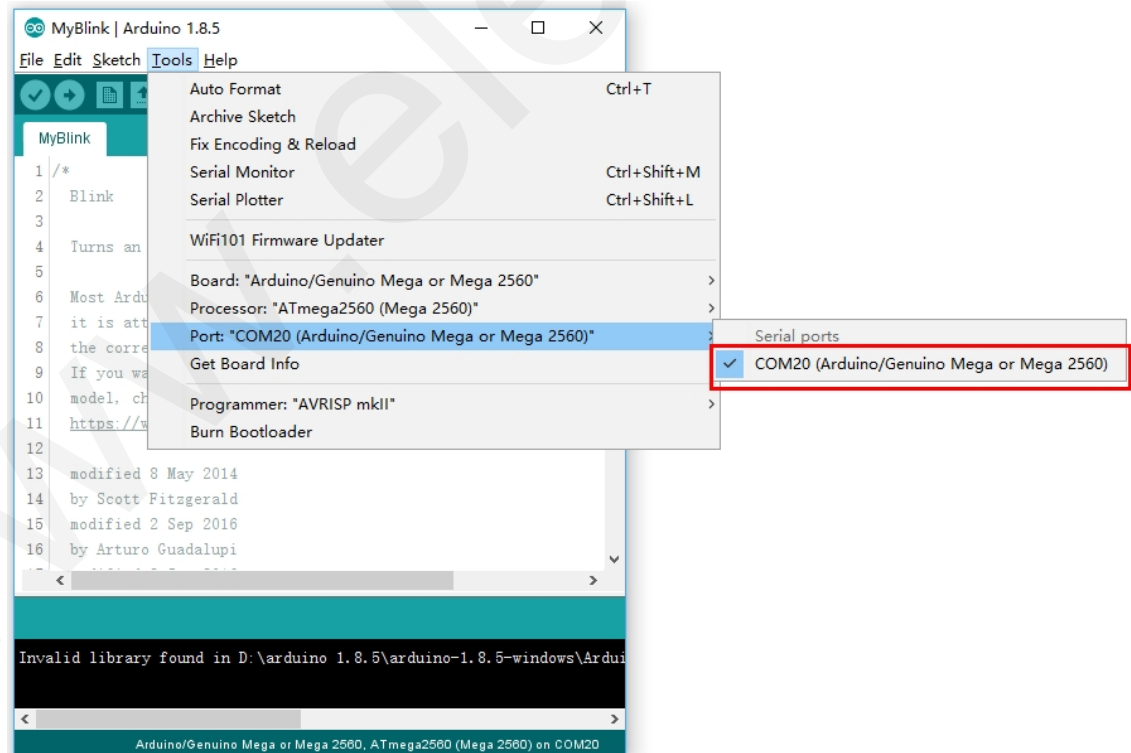
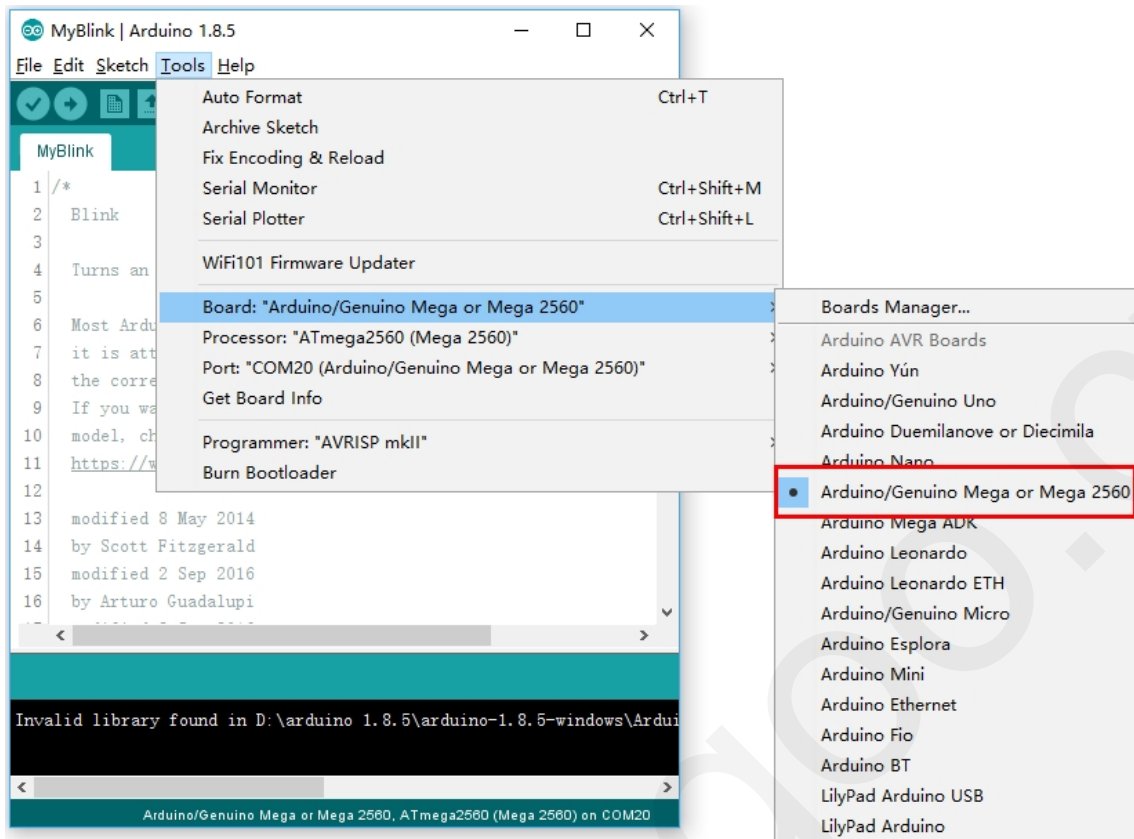




Uložili jste si do skicáku kopii obrázku "Blink". To znamená, že pokud ji budete chtít znovu najít, stačí ji otevřít pomocí nabídky Soubor > Skicák.

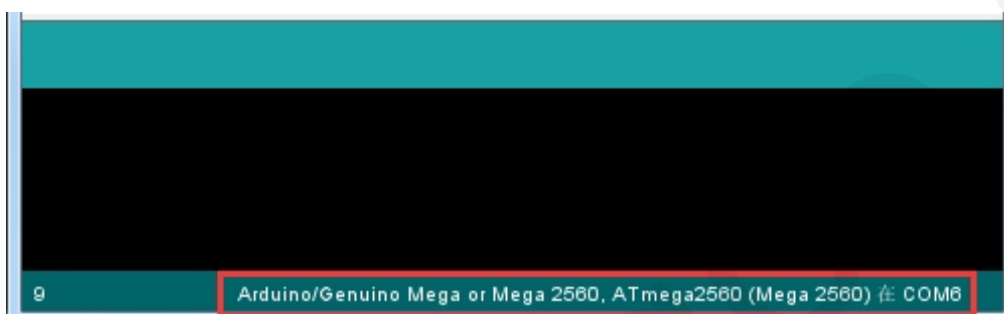


Připojte desku Arduino k počítači pomocí kabelu USB a zkontrolujte, zda jsou správně nastaveny položky "Board Type" a "Serial Port".



Poznámka: Typ desky a sériový port zde nemusí odpovídat obrázku. Pokud používáte 2560, pak musíte jako typ desky zvolit Mega 2560, ostatní volby lze provést stejným způsobem. A zobrazený Sériový port je pro každého jiný, přestože je zde zvolen COM 20, může to být COM3 nebo COM4 ve vašem počítači. Správný COM port má být COMX (arduino XXX), což je podle certifikačních kritérií.

Arduino IDE vám v dolní části okna zobrazí aktuální nastavení desky.



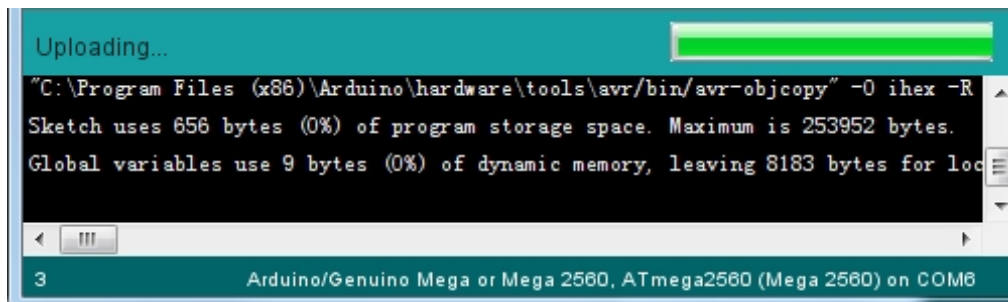
Klikněte na tlačítko "Nahrát". Druhé tlačítko zleva na panelu nástrojů.



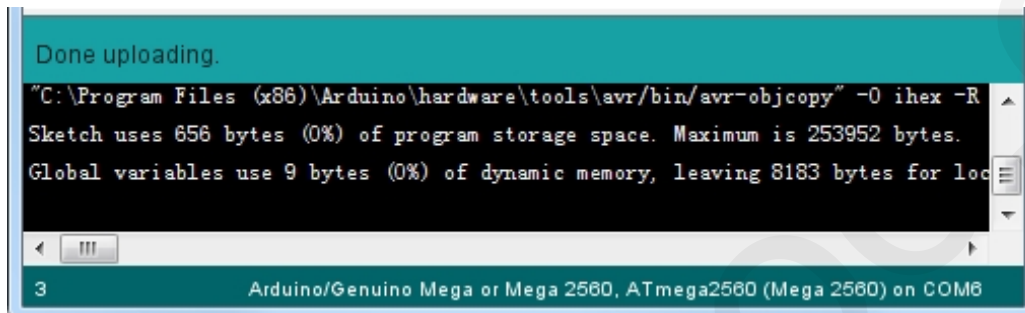
Pokud sledujete stavovou oblast IDE, uvidíte ukazatel průběhu a řadu zpráv. Nejprve se zobrazí zpráva "Compiling Sketch...". Tím se skica převede do formátu vhodného pro nahrání na desku.



Poté se stav změní na "Nahrávání". V tomto okamžiku by měly LED diody na Arduino začít blikat, protože se skica přenáší.



Nakonec se stav změní na "Hotovo".



Druhá zpráva nám říká, že sketch využívá 928 bajtů z dostupných 32 256 bajtů. Po fázi "Compiling Sketch.." se může zobrazit následující chybové hlášení:



Může to znamenat, že vaše deska není vůbec připojena, nebo že nebyly nainstalovány ovladače (pokud je to nutné), nebo že je vybrán nesprávný sériový port.

Pokud se s tím setkáte, vraťte se k lekci 0 a zkontrolujte instalaci.

Po dokončení nahrávání by se deska měla restartovat a začít blikat.

Otevřete kód

Všimněte si, že velkou část tohoto náčrtu tvoří komentáře. Nejsou to skutečné instrukce programu, ale spíše jen vysvětlují, jak program funguje. Jsou zde pro váš užitek.

Vše mezi /* a */ v horní části náčrtu je blokový komentář; vysvětluje, k čemu náčrt slouží.

Jednořádkové komentáře začínají znakem `//` a vše až do konce tohoto řádku je považováno za komentář.

První řádek kódu je:

```
int led = 13;
```

Jak je vysvětleno v komentáři výše, jedná se o pojmenování pinu, ke kterému je LED připojena. Na většině Arduin, včetně MEGA 2560 a Leonardo, je to 13. Dále tu máme funkci "setup". Opět, jak je uvedeno v komentáři, tato funkce se provede po stisknutí tlačítka reset. Provede se také vždy, když se deska z jakéhokoli důvodu resetuje, například když je poprvé připojena k napájení nebo po nahrání náčrtu.

```
void setup() {  
  // inicializujte digitální pin jako výstup.
```

```
  pinMode (led, OUTPUT);  
}
```

Každý náčrt Arduina musí mít funkci "setup" a místo, kam byste mohli chtít přidat vlastní instrukce, je mezi { a }.

V tomto případě je zde pouze jeden příkaz, který, jak je uvedeno v komentáři, říká desce Arduino, že budeme používat pin LED jako výstup.

Povinnou součástí náčrtu je také funkce 'loop'. Na rozdíl od funkce "setup", která se po resetu spustí pouze jednou, se funkce "loop" po dokončení příkazů okamžitě spustí znovu.

```
void loop() {  
  digitalWrite(led,);           //zapnutí LED (HIGH je úroveň napětí)  
  delay(1000);                  // počkejte sekundu  
  digitalWrite(led,);           //vypněte LED diodu tak, že napětí bude LOW  
  delay(1000);                  // počkejte sekundu.  
}
```

Uvnitř funkce smyčky příkazy nejprve zapnou pin LED (HIGH), pak "zpoždění" na 1000 milisekund (1 sekunda), pak pin LED vypnou a na další sekundu pozastaví.

Nyní přimějete LED diodu blikat rychleji. Jak jste možná uhodli, klíč k tomu spočívá ve změně parametru v () pro příkaz 'delay'.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the volt
33   delay(500)                        // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the vo
35   delay(500)                        // wait for a second
36 }
```

Tato doba zpoždění se udává v milisekundách, takže pokud chcete, aby LED blikala dvakrát rychleji, změňte hodnotu z 1000 na 500. Pak by se každé zpoždění pozastavilo na půl sekundy, nikoli na celou sekundu.

Znovu nahrajte náčrtek a měli byste vidět, že LED dioda začne rychleji blikat.

Lekce 3 LED

Přehled

V této lekci se naučíte měnit jas LED pomocí různých hodnot rezistoru.

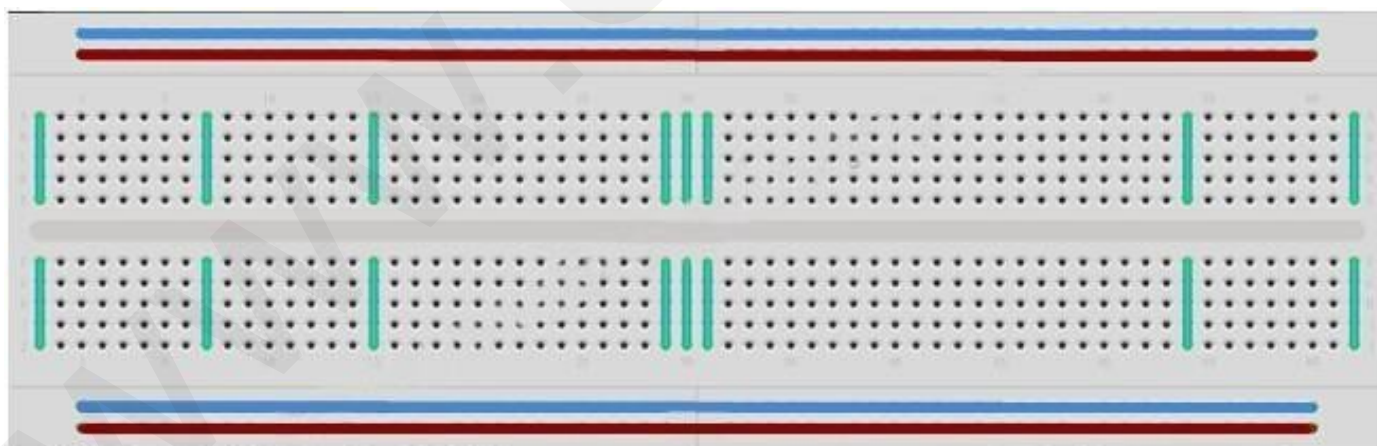
Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x 5 mm červená LED dioda
- (1) x 220 ohmů
- (1) x 1k ohmový rezistor
- (1) x 10k ohmový odpor
- (2) x vodiče M-M (propojovací vodiče typu Male to Male)

Úvod do složky

CHLEBOVÁ DESKA MB-102 :

Deska umožňuje rychlé vytváření prototypů obvodů bez nutnosti pájení spojů. Níže je uveden příklad.



Breadboardy se dodávají v různých velikostech a konfiguracích. Nejjednodušší typ je jen mřížka otvorů v plastovém bloku. Uvnitř jsou kovové proužky, které zajišťují elektrické spojení mezi otvory v kratších řadách. Zasunutím nožiček dvou různých součástek do stejné řady se elektricky spojí. Hluboký kanálek probíhající uprostřed naznačuje, že je zde přerušení spojení, což znamená, že můžete zasunout čip s nožičkami na obou stranách kanálu, aniž by se spojily dohromady. Některé breadboardy mají podél dlouhých okrajů desky dva pruhy otvorů, které jsou odděleny od hlavní mřížky. Ty mají proužky probíhající po celé délce desky uvnitř a poskytují možnost připojení společného napětí. Obvykle jsou v párech pro +5 V a zem. Tyto proužky se označují jako lišty a umožňují připojit napájení k mnoha součástkám nebo bodům na desce.

Přestože jsou destičky skvělé pro tvorbu prototypů, mají určitá omezení. Vzhledem k tomu, že se jedná o dočasné spoje, nejsou tak spolehlivé jako pájené spoje. Pokud máte s nějakým obvodem občasné problémy, může to být způsobeno špatným připojením na desce.

LED DIODA:

LED diody jsou skvělými kontrolkami. Spotřebovávají velmi málo elektřiny a vydrží prakticky věčně.

V této lekci použijete asi nejběžnější LED diodu: 5 mm červenou LED diodu. Průměr 5 mm znamená průměr LED diody. Další běžné velikosti jsou 3 mm a 10 mm. LED diodu nemůžete připojit přímo k baterii nebo zdroji napětí, protože 1) LED dioda má kladný a záporný vodič a nebude svítit, pokud bude umístěna špatně a 2) LED dioda musí být použita s rezistorem, který omezí nebo "přiškrtní" množství proudu, který jí protéká; jinak se spálí!



Pokud u LED diody nepoužijete rezistor, může dojít k jejímu téměř okamžitému zničení, protože jí bude protékat příliš velký proud, který ji zahřeje a zničí "spoj", na kterém vzniká světlo.

Existují dva způsoby, jak zjistit, který vodič LED je kladný a který záporný.

Za prvé, kladný náskok jedlounodobější.

Za druhé, v místě, kde záporný vodič vstupuje do těla LED, je plochá hrana pouzdra LED.

Pokud máte LED diodu, která má plochou stranu vedle delšího vodiče, měli byste předpokládat, že delší vodič je kladný.

ODPORY:

Jak již název napovídá, odpory kladou odpor toku elektrické energie. Čím vyšší je hodnota rezistoru, tím větší odpor klade a tím méně elektrického proudu jím protéká. Toho využijeme k tomu, abychom mohli řídit, kolik elektřiny LED diodou protéká, a tedy jak jasně svítí.

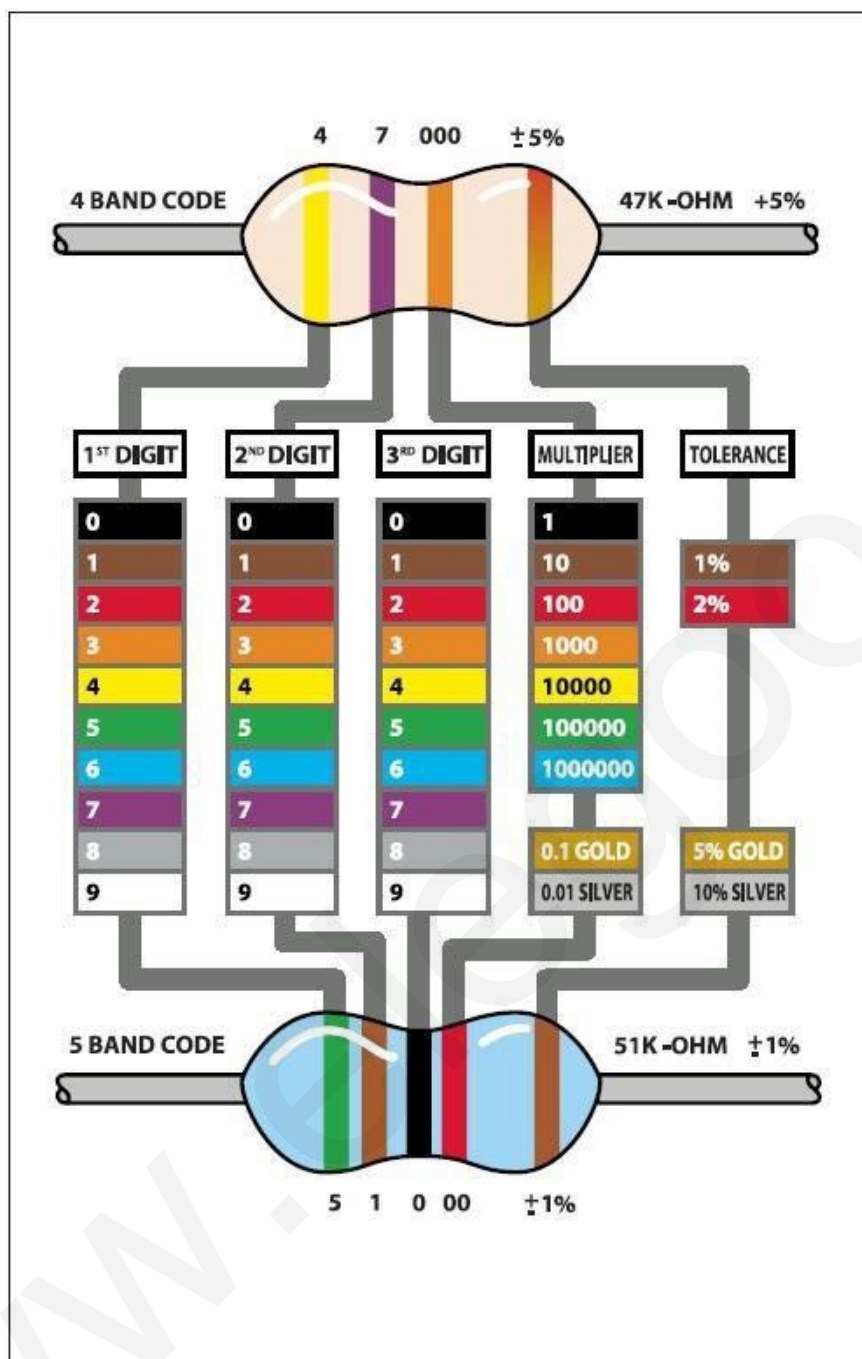


Nejprve však více odporech...

Jednotka odporu se nazývá ohm, který se obvykle zkracuje na Ω řecké písmeno Omega. Protože ohm je nízká hodnota odporu (neklade vůbec žádný velký odpor), označujeme hodnoty rezistorů také v $k\Omega$ (1 000 Ω) a $M\Omega$ (1 000 000 Ω). Ty se nazývají kiloohmy a megaohmy.

V této lekci použijeme tři různé hodnoty rezistoru: 220Ω , $1k\Omega$ a $10k\Omega$. Všechny tyto rezistory vypadají stejně, jen mají na sobě různé barevné proužky. Tyto proužky vám řeknou, jakou hodnotu má daný rezistor.

Barevné označení rezistoru má tři barevné pruhy a na jednom konci zlatý pruh.



Na rozdíl od LED nemají rezistory kladný a záporný vodič. Mohou být připojeny oběma směry.

Pokud se vám tato metoda zdá příliš složitá, můžete si na našich rezistorech přečíst přímo barevný kroužek a určit hodnotu odporu. Nebo můžete místo toho použít digitální multimetr.

Připojení

Schéma

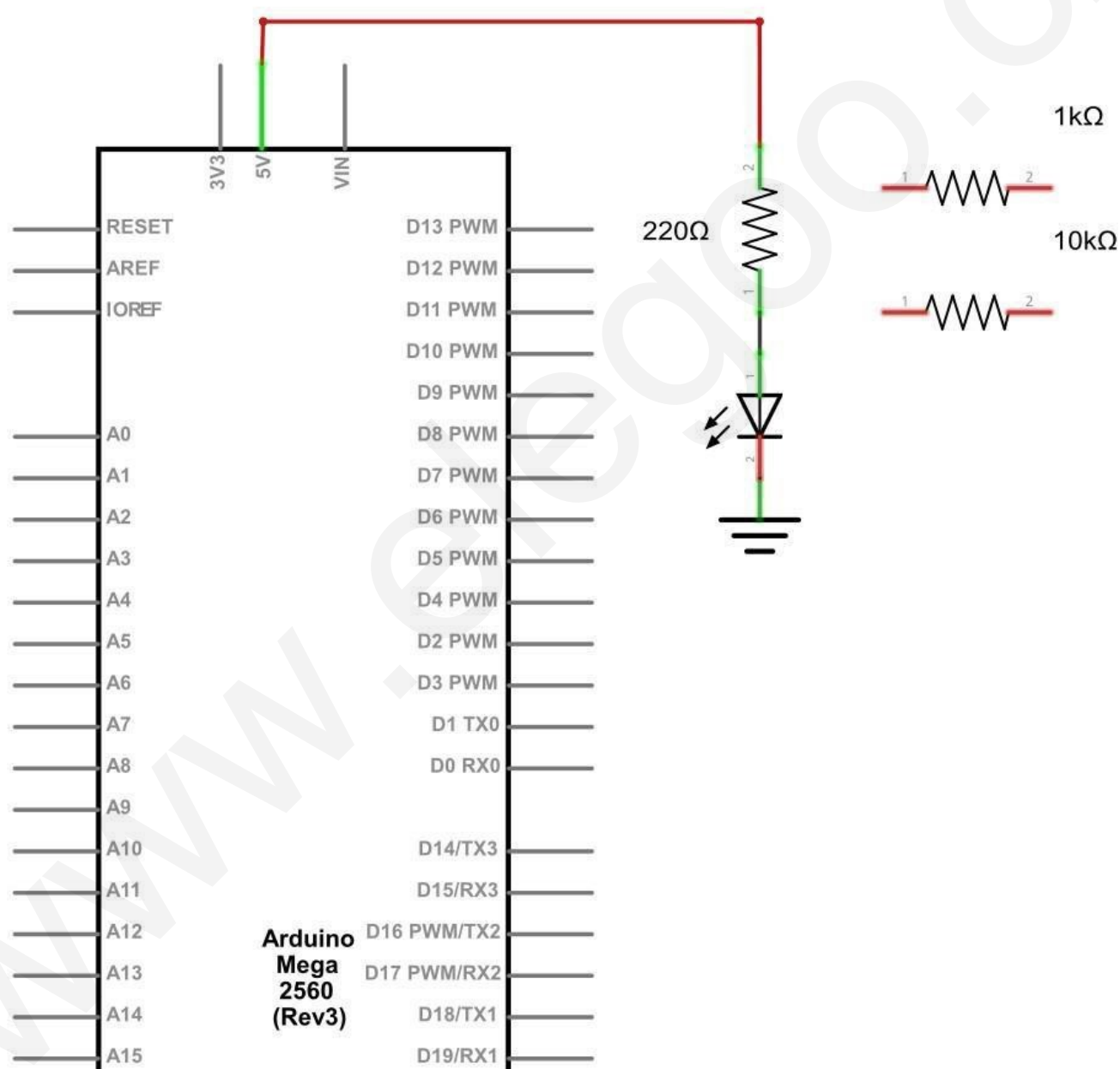
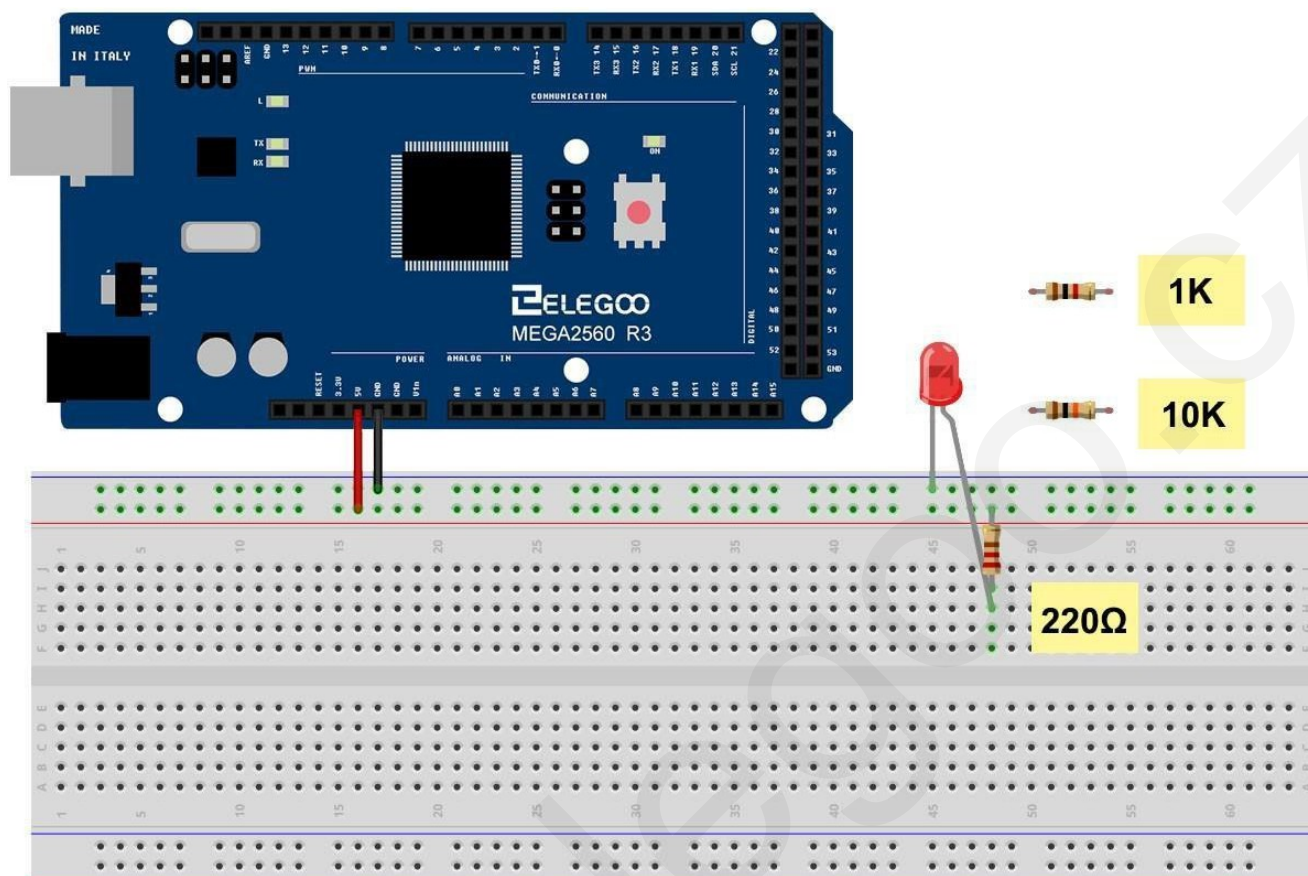


Schéma zapojení



MEGA 2560 je vhodným zdrojem 5 V, které použijeme k napájení LED a rezistoru. Se zdrojem MEGA 2560 nemusíte dělat nic jiného, než jej připojit ke kabelu USB.

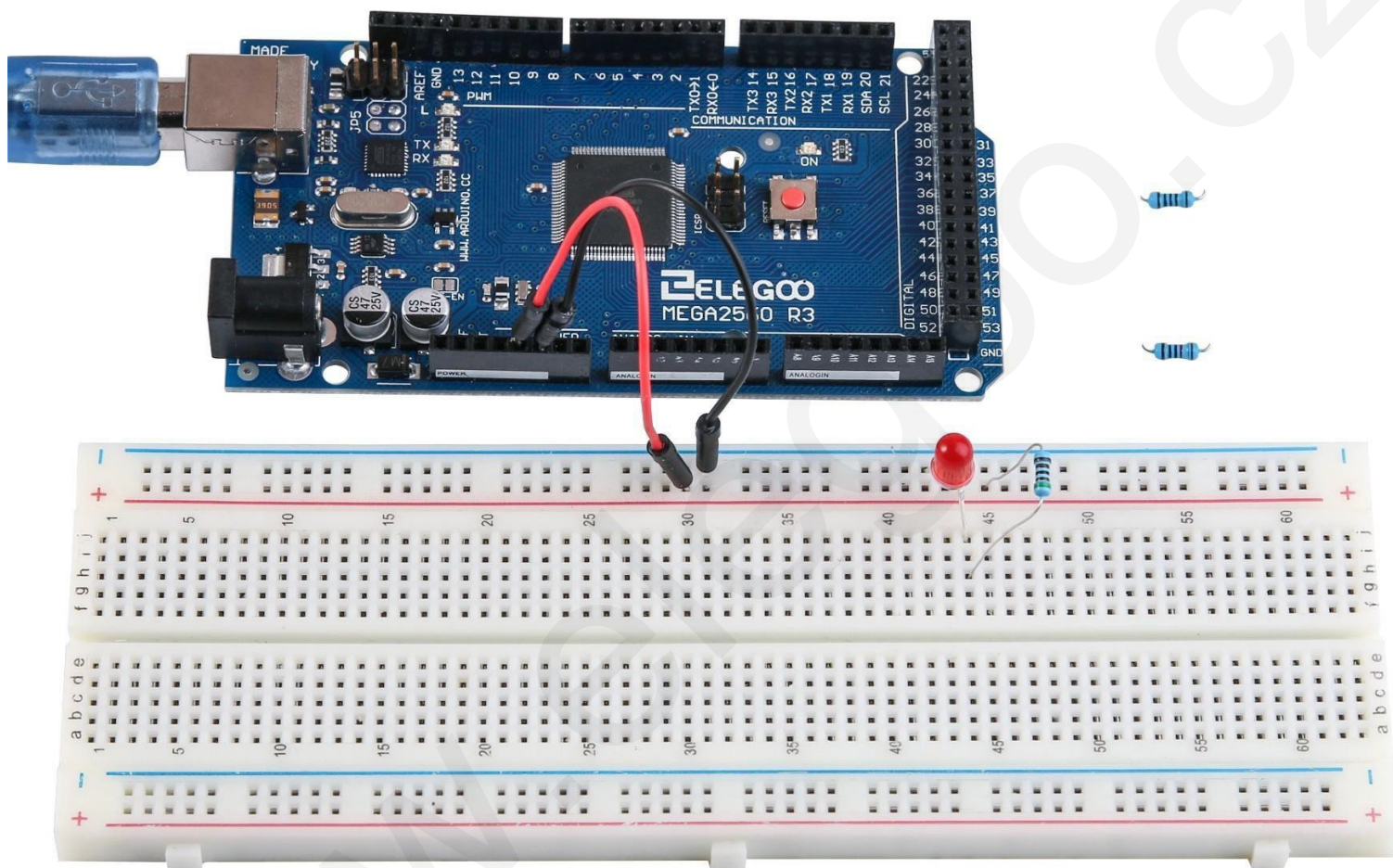
S rezistorem 220 Ω na místě by měla být LED dioda poměrně jasná. Pokud vyměníte rezistor 220 Ω za rezistor 1 k Ω , bude LED dioda o něco slabší. A konečně, s rezistorem 10 k Ω na místě bude LED dioda téměř viditelná. Vytáhněte červený vodič propojky z chlebníku a dotkněte se ho v otvoru a vyjměte ho, aby se choval jako spínač. Rozdíl by měl být právě patrný.

V současné době máte 5 V na jedné noze rezistoru, druhá noha rezistoru jde na kladnou stranu LED a druhá strana LED jde na GND. Pokud bychom však rezistor přesunuli tak, aby byl za LED diodou, jak je znázorněno níže, LED dioda bude stále svítit.

Pravděpodobně budete chtít vrátit na místo 220 Ω rezistor.

Nezáleží na tom, na kterou stranu LED diody rezistor umístíme, pokud tam někde je.

Příklad obrázek



Lekce 4 RGB LED

Přehled

LED diody RGB představují zábavný a snadný způsob, jak přidat do svých projektů trochu barev. Protože jsou jako 3 běžné LED diody v jedné, způsob jejich použití a zapojení se příliš neliší.

Většinou se dodávají ve 2 verzích: Varianty: se společnou anodou nebo se společnou katodou. Common Anode používá 5V na společném pinu, zatímco Common Cathode se připojuje k zemi.

Stejně jako u každé LED diody musíme zapojit několik rezistorů (celkem 3), abychom mohli omezit odebíraný proud.

V našem náčrtu začneme s LED diodou v červené barvě, poté přejdeme do zelené, pak do modré a nakonec zpět do červené barvy. Tímto postupem projdeme většinu barev, kterých lze dosáhnout.

Požadovaná součást:

- (1) x Elegoo Mega2560 R3
- (1) x 830 vázacích bodůDeska na chléb
- (4) x vodiče M-M (propojovací vodiče)
- (1) x RGB LED
- (3) x 220 ohmů rezistory

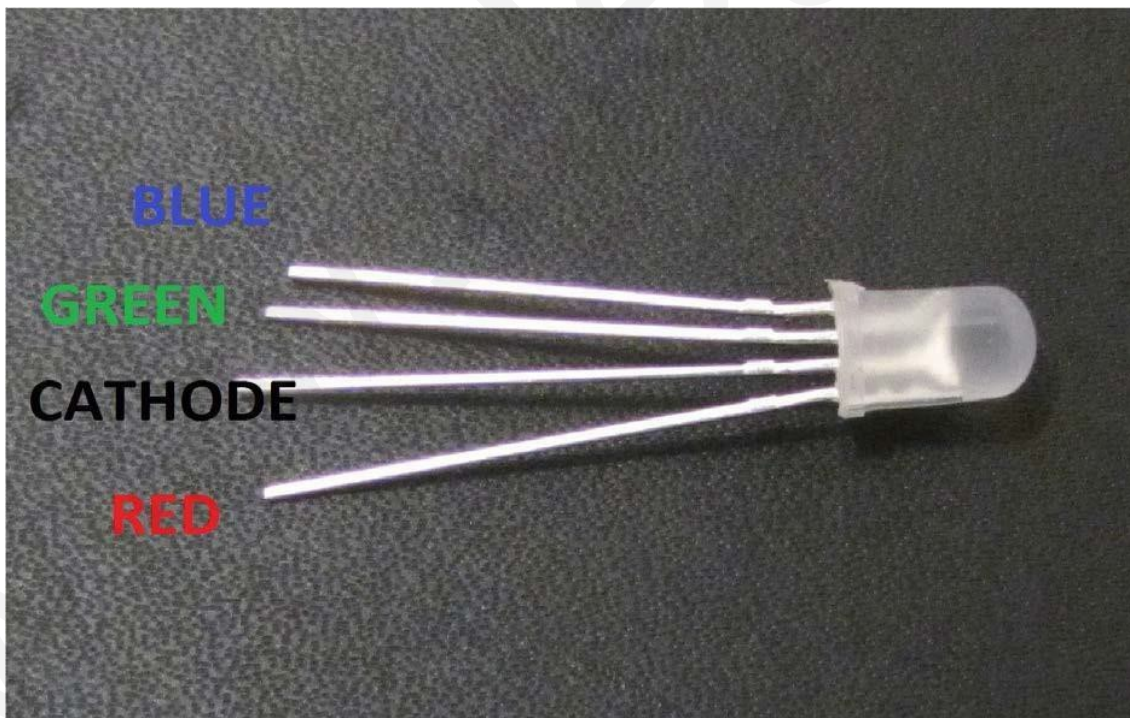
Úvod do složky

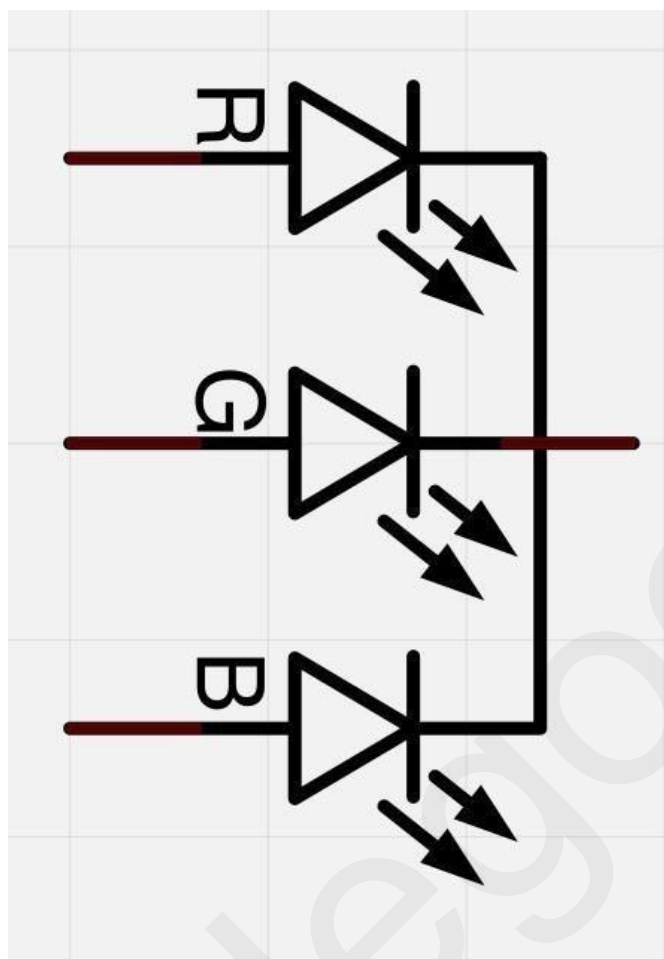
RGB:

LED diody RGB (červená, zelená a modrá) vypadají na první pohled stejně jako běžné LED diody. Uvnitř běžného balení LED diod se však ve skutečnosti nacházejí tři LED diody, jedna červená, jedna zelená a ano, jedna modrá. Ovládáním jasu jednotlivých diod LED můžete namíchat prakticky jakoukoli barvu.

Barvy mícháme stejným způsobem, jako byste míchali barvy na paletě - nastavením jasu každé ze tří diod LED. Nejsložitější způsob, jak to udělat, by bylo použít rezistory různých hodnot (nebo proměnné rezistory), jak jsme to udělali v lekci 2, ale to je spousta práce! Naštěstí pro nás má deska MEGA 2560 R3 analogovou funkci Write, kterou můžete použít s piny označenými symbolem ~ k vyvedení proměnného množství energie do příslušných LED diod.

LED dioda RGB má čtyři vodiče. Jeden vodič vede ke kladnému připojení každé z jednotlivých LED v balení a jeden vodič je připojen ke všem třem záporným stranám LED.





Zde na fotografiích vidíte 4 elektrody LED. Každý samostatný vývod pro zelenou, modrou nebo červenou barvu se nazývá anoda. Vždy k němu připojíte "+". Katoda se připojuje na "-" (zem). Pokud ji připojíte opačně, LED dioda nebude svítit.

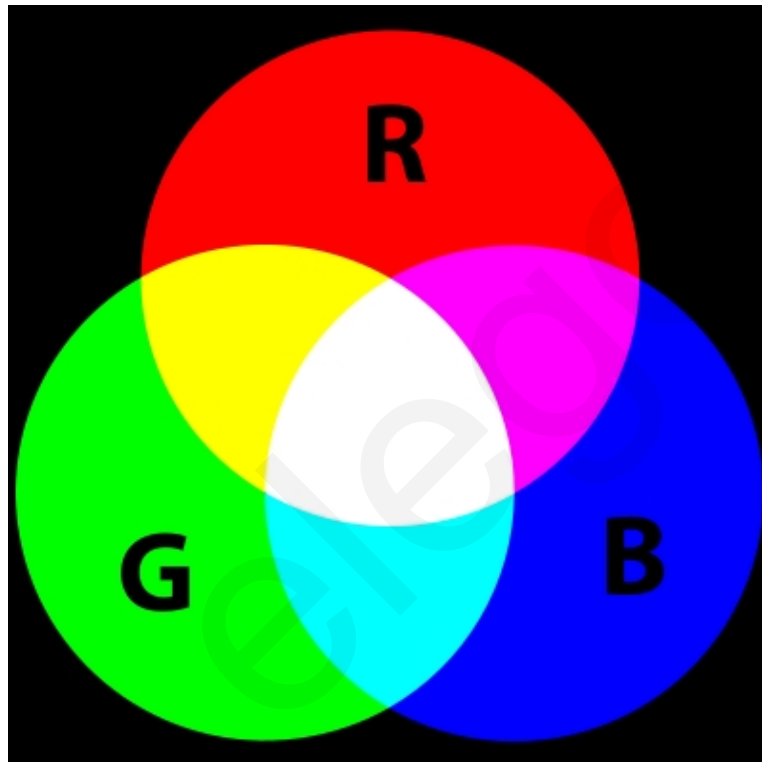
Společným záporným připojením pouzdra LED je druhý pin zploché strany. Je také nejdelší ze čtyř vodičů a bude připojen k zemi.

Každá LED dioda uvnitř pouzdra vyžaduje vlastní 220Ω rezistor, aby jí neprotékal příliš velký proud. Tři kladné vodiče LED diod (jeden červený, jeden zelený a jeden modrý) jsou pomocí těchto rezistorů připojeny k výstupním pinům MEGA 2560.

BARVA: ZELENÁ, ZELENÁ, ZELENÁ, ZELENÁ, ZELENÁ, ZELENÁ, ZELENÁ:

Důvodem, proč si můžete namíchat jakoukoli barvu, když měníte množství červeného, zeleného a modrého světla, je to, že vaše oko má tři typy světelných receptorů (červený, zelený a modrý). Vaše oko a mozek zpracovávají množství červeného, zeleného a modrého světla a převádějí je na barvu spektra.

Použitím tří LED diod si svým způsobem hrajeme s okem. Stejná myšlenka se používá v televizorech, kde jsou na LCD displeji červené, zelené a modré barevné body vedle sebe a tvoří každý pixel.



Pokud nastavíme stejný jas všech tří diod LED, bude celková barva světla bílá. Pokud vypneme modrou LED diodu, takže stejný jas budou mít pouze červená a zelená LED dioda, bude světlo vypadat žlutě.

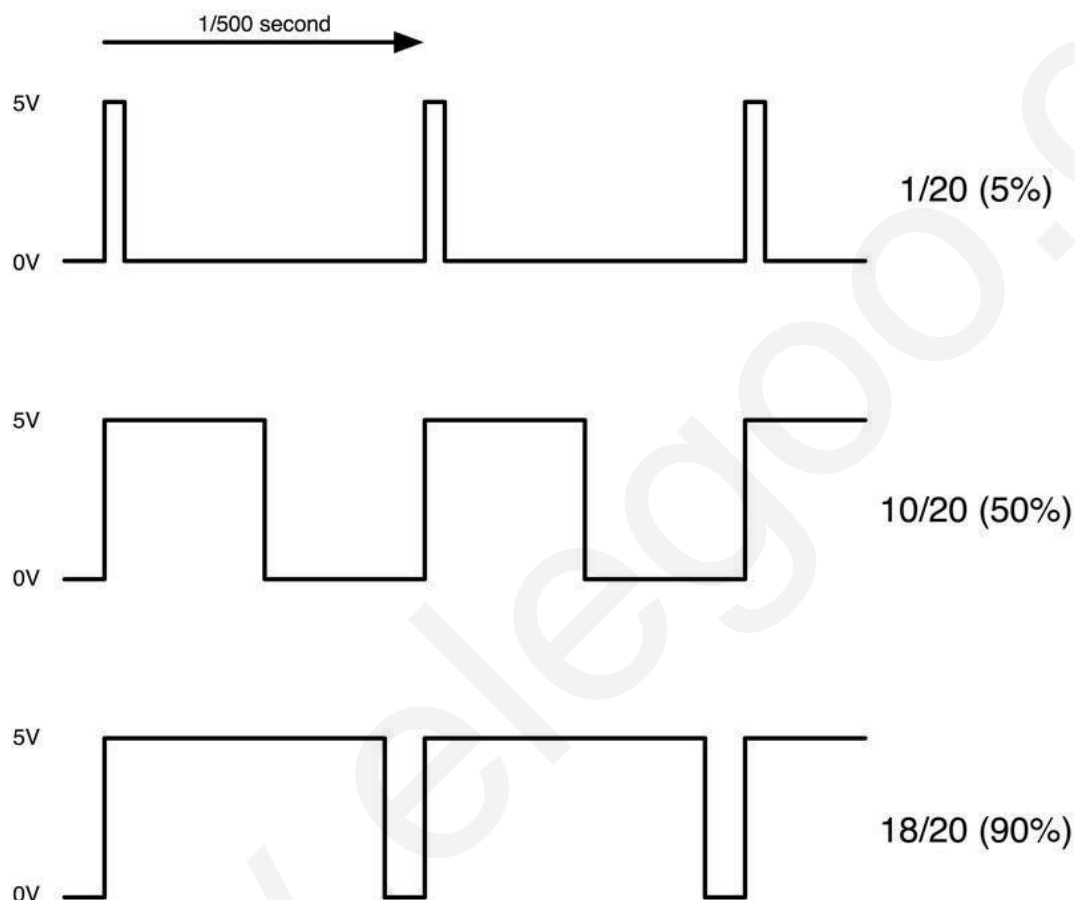
Jas každé z červených, zelených a modrých částí diody LED můžeme ovládat samostatně, takže je možné namíchat libovolnou barvu.

Černá není ani tak barva, jako spíše nepřítomnost světla. Nejblíže černé barvě se proto můžeme s naší LED diodou přiblížit tak, že vypneme všechny tři barvy.

Teorie (PWM)

Pulzně šířková modulace (PWM) je technika řízení výkonu. Zde ji používáme také k řízení jasu jednotlivých LED diod.

Níže uvedené schéma ukazuje signál z jednoho z pinů PWM na MEGA 2560.



Zhruba každou 1/500 sekundy se na výstupu PWM objeví puls. Délka tohoto impulsu se řídí funkcí "analog Write". Funkce 'analog Write(0)' tedy nevytvoří vůbec žádný impuls a funkce 'analog Write(255)' vytvoří impuls, který trvá až do příštího impulsu, takže výstup je vlastně zapnutý po celou dobu. Pokud v analogovém zápisu zadáme hodnotu, která je někde mezi 0 a 255, pak vytvoříme impuls. Pokud je výstupní impuls vysoký pouze po dobu 5 % času, pak to, co řídíme, obdrží pouze 5 % plného výkonu.

Pokud je však na výstupu po 90 % času napětí 5 V, pak zátěž dostane 90 % energie, která je jí dodávána. Při této rychlosti nevidíme, jak se LED diody zapínají a vypínají, takže nám to připadá, jako by se jen měnil jas.

Připojení

Schéma

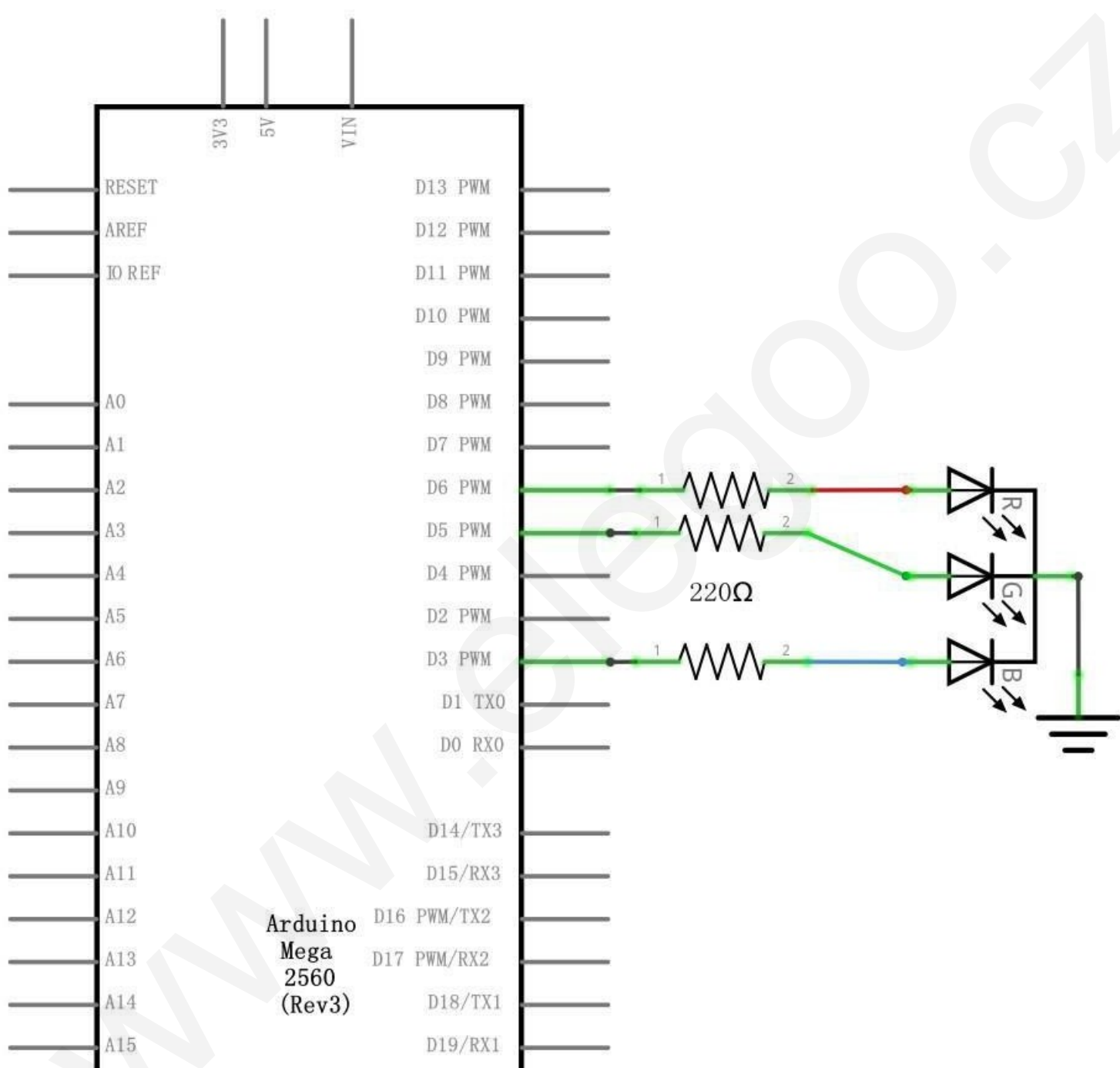
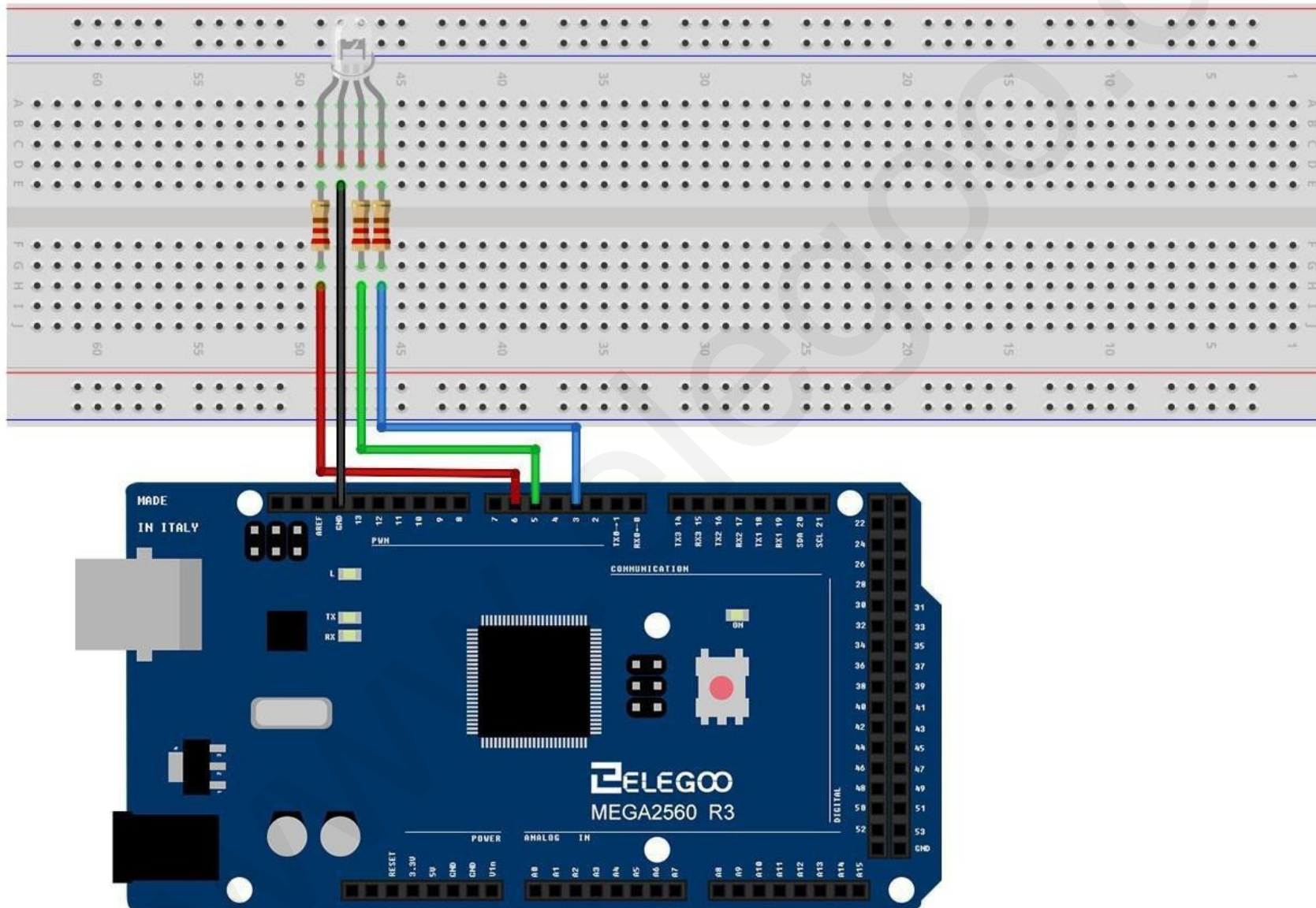
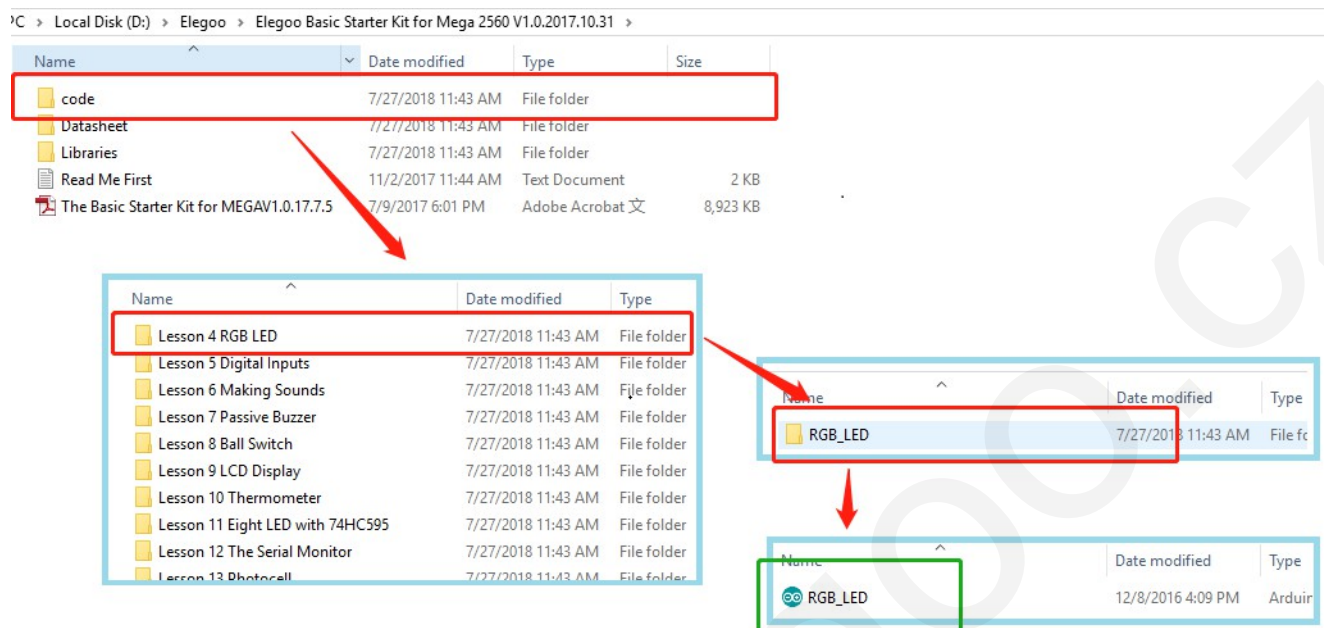


Schéma zapojení



Kód

Po zapojení otevřete Sketch v cestě ke složce: Klepněte na tlačítko UPLOAD a nahrajte program.



Podrobnosti o nahrávání programu v případě chyb naleznete v lekcí 2.

Náčrt začíná určením, které piny budou použity pro jednotlivé barvy:

```
// Definice pinů
```

```
#define BLUE 3
```

```
#define GREEN 5
```

```
#define RED 6
```

Dalším krokem je napsání funkce 'setup'. Jak jsme se naučili v předchozích lekcích, funkce setup se spustí pouze jednou po resetu Arduina. V tomto případě stačí definovat tři piny, které používáme jako výstupy.

```
void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
}
```

Než se podíváme na funkci 'loop', podívejme se na poslední funkci v náčrtu.

Definiční proměnné

```
redValue = 255; // zvolte hodnotu mezi 1 a 255 pro změnu barvy.
```

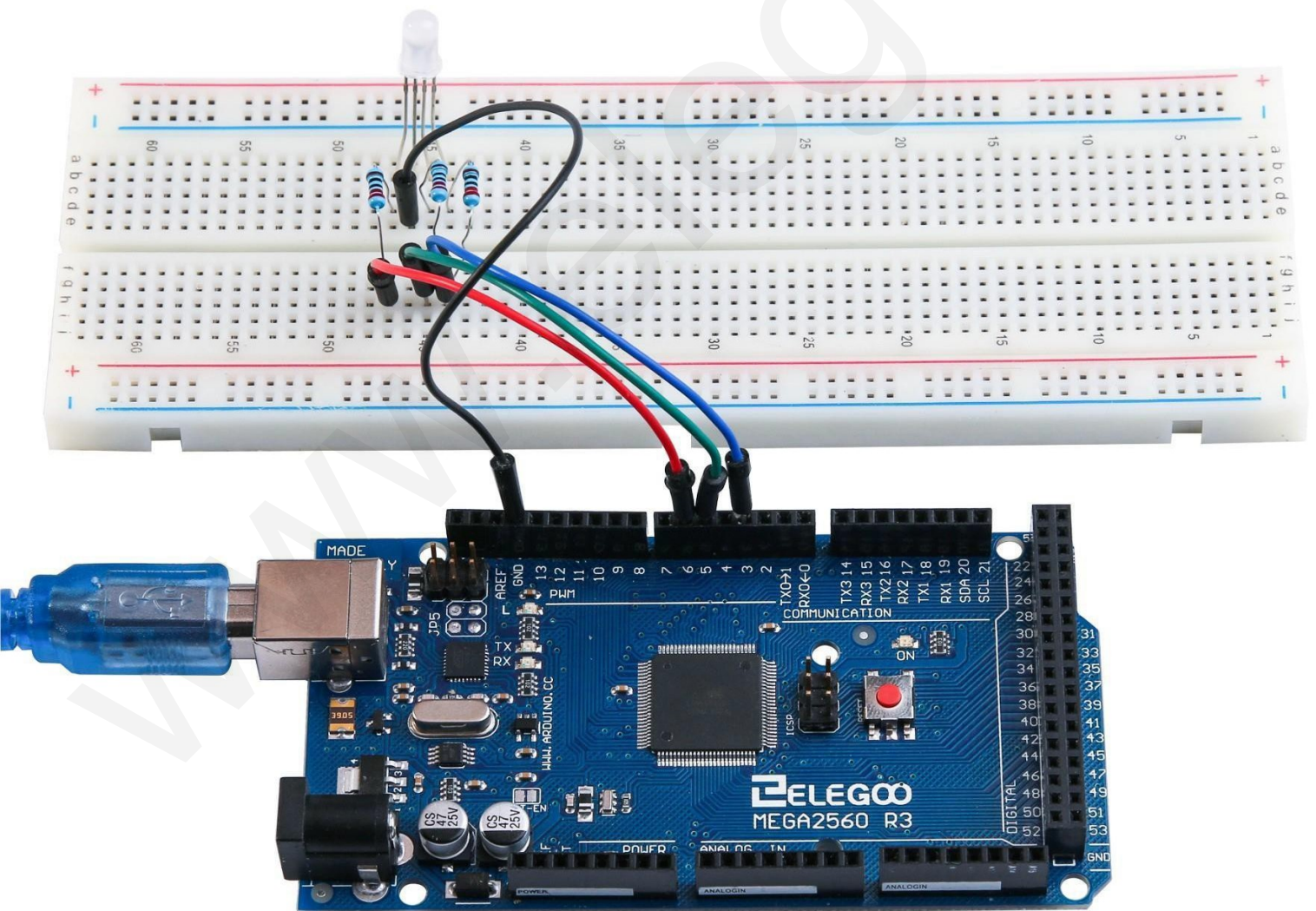
```
greenValue = 0;
```

```
blueValue = 0;
```

Tato funkce přijímá tři argumenty, jeden pro jas červené, zelené a modré LED diody. V každém případě bude číslo v rozsahu 0 až 255, kde 0 znamená vypnuto a 255 znamená maximální jas. Funkce pak zavolá 'analogWrite', aby nastavila jas každé LED.

Zkuste do náčrtu přidat několik vlastních barev a sledujte, jaký efekt bude mít LED dioda.

Příklad obrázku



Lekce 5 Digitální vstupy

Přehled

V této lekci se naučíte používat tlačítka s digitálními vstupy k zapínání a vypínání LED.

Stisknutím tlačítka se LED dioda zapne, stisknutím druhého tlačítka se LED dioda vypne.

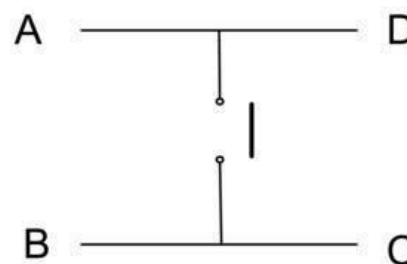
Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x 830 vázacích bodů Deska na chleba
- (1) x 5 mm červená LED dioda
- (1) x 220 ohmů odpor
- (2) x tlačítkové spínače
- (7) x vodiče M-M (propojovací vodiče typu Male to Male)

Úvod do složky

STLAČOVACÍ PŘEPÍNAČE:

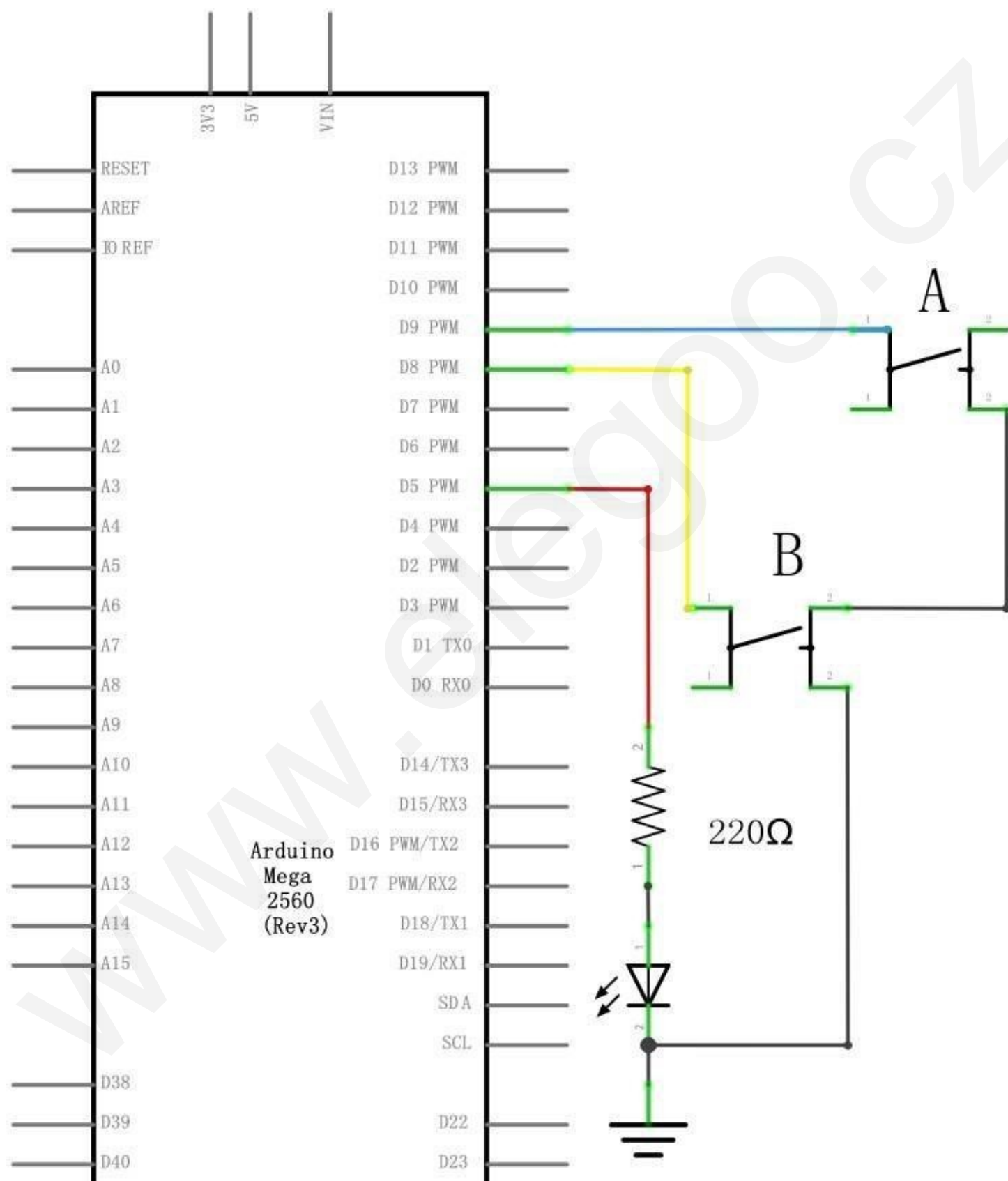
Přepínače jsou opravdu jednoduché komponenty. Když stisknete tlačítko nebo otočíte páčku, spojí dva kontakty dohromady, aby jimi mohla protékat elektřina. Malé hmatové spínače, které se používají v této lekci, mají čtyři připojení, což může být trochu matoucí.



Ve skutečnosti existují pouze dvě elektrická připojení. Uvnitř spínacího pouzdra jsou k sobě připojeny vývody B a C, stejně jako A a D.

Připojení

Schéma



Přestože jsou těla spínačů čtvercová, kolíky vyčnívají z opačných stran spínače. To znamená, že kolíky budou dostatečně daleko od sebe pouze tehdy, když budou správně umístěny na desce.

Nezapomeňte, že LED dioda musí mít kratší záporný vodič vlevo.

Kód

Po zapojení otevřete program ve složce s kódem - Lekce 5 Digitální vstupy a stiskněte **UPLOAD** pro nahrání programu. Pokud se zobrazí chybové hlášení, podívejte se do Lekce 2, kde najdete podrobnosti o návodu na nahrání programu.

Načtete náčrt na desku MEGA2560. Stisknutím levého tlačítka LED diodu zapnete, zatímco stisknutím pravého tlačítka ji vypnete.

V první části náčrtu jsou definovány tři proměnné pro tři piny, které mají být použity. 'ledPin' je výstupní pin a 'buttonApin' se bude vztahovat na spínač blíže k horní části desky a 'buttonBpin' na druhý spínač.

Funkce 'setup' definuje ledPin jako OUTPUT jako obvykle, ale nyní máme dva vstupy, se kterými se musíme vypořádat. V tomto případě nastavíme pinMode na 'INPUT_PULLUP' takto:

```
pinMode(buttonApin, INPUT_PULLUP);  
pinMode(buttonBpin, INPUT_PULLUP);
```

Režim pinu INPUT_PULLUP znamená, že pin má být používán jako vstup, ale pokud k němu není připojeno nic jiného, měl by být "vytažen" na HIGH. Jinými slovy, výchozí hodnota pro vstup je HIGH, pokud není vytažen LOW stisknutím tlačítka.

Proto jsou přepínače připojeny ke GND. Když je spínač stisknut, připojí vstupní pin na GND, takže již není HIGH.

Vzhledem k tomu, že vstup je normálně HIGH a přechází na LOW pouze při stisknutí tlačítka, je logika trochu převrácená. To budeme řešit ve funkci 'loop'.

```
void loop()  
{  
  if (digitalRead(buttonApin) == LOW)  
  {  
    digitalWrite(ledPin, HIGH);  
  }  
  if (digitalRead(buttonBpin) == LOW)
```

```

{
    digitalWrite(ledPin, LOW);
}
}

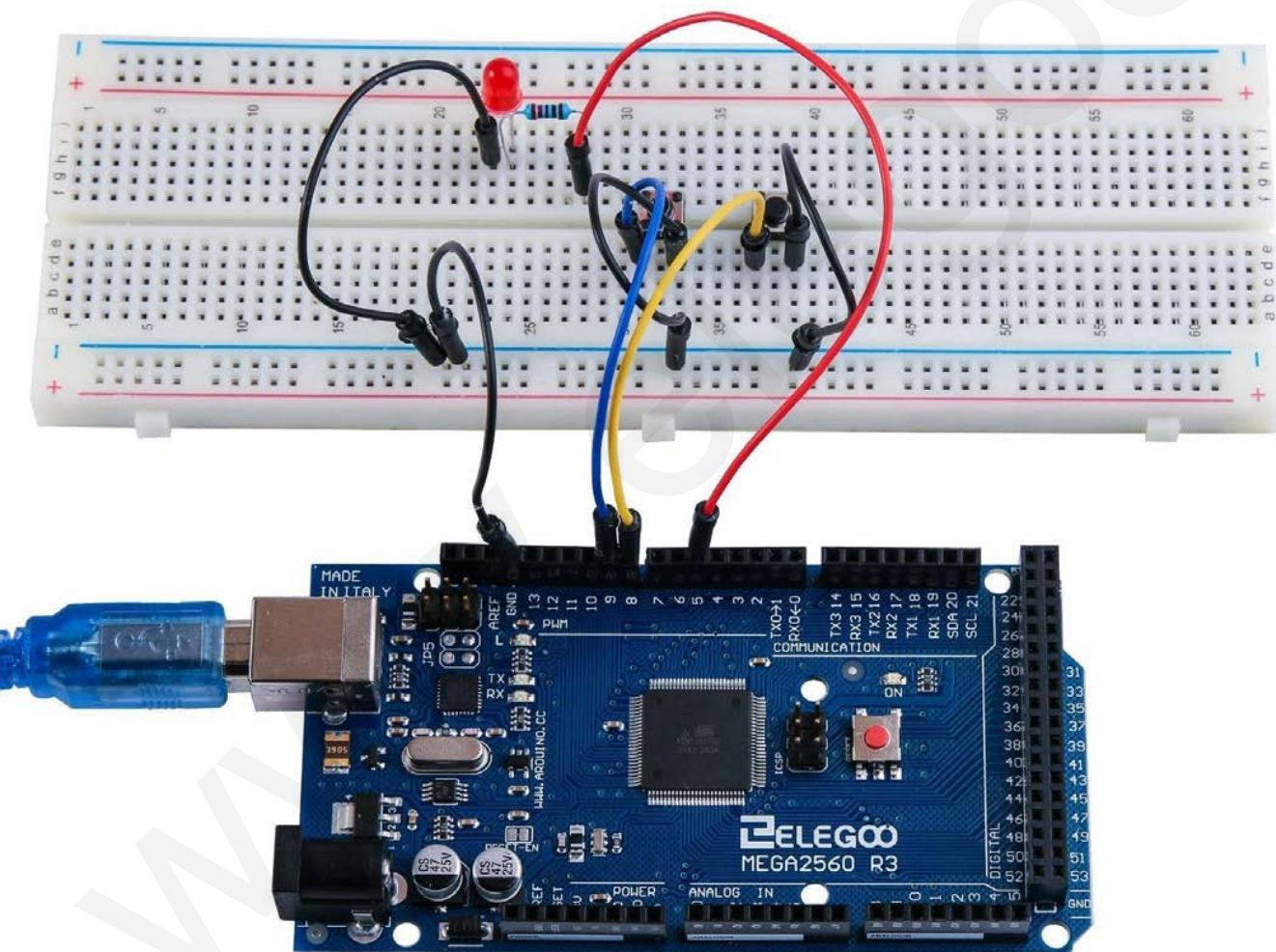
```

Ve funkci 'loop' jsou dva příkazy 'if'. Pro každé tlačítko jeden. Každý z nich provede 'digitalRead' na příslušném vstupu.

Nezapomeňte, že pokud je tlačítko stisknuto, příslušný vstup bude LOW, pokud je tlačítko A nízké, pak ho zapne 'digitalWrite' na ledPin.

Podobně, pokud je stisknuto tlačítko B, je na ledPin zapsána hodnota LOW.

Příklad obrázku



Lekce 6 Aktivní bzučák

Přehled

V této lekci se dozvíte, jak generovat zvuk pomocí aktivního bzučáku.

Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x Aktivní bzučák
- (2) x vodiče F-M (samice na samce DuPont)

Úvod do složky

BZUČÁK:

Elektronické bzučáky jsou napájeny stejnosměrným proudem a jsou vybaveny integrovaným obvodem. Jsou široce používány v počítačích, tiskárnách, kopírkách, alarmech, elektronických hračkách, elektronických zařízeních pro automobily, telefonech, časovačích a dalších elektronických výrobcích pro hlasová zařízení. Bzučáky lze rozdělit na aktivní a pasivní. Otočte vývody dvou bzučáků směrem nahoru. Ten se zeleným plošným spojem je pasivní bzučák, zatímco druhý uzavřený černou páskou je aktivní.

Rozdíl mezi nimi spočívá v tom, že aktivní bzučák má zabudovaný oscilační motor. zdroj, takže při elektrifikaci vydává zvuk. Pasivní bzučák takový zdroj nemá, takže při použití stejnosměrných signálů nebude bzučet; místo toho je třeba k jeho buzení použít čtvercové vlny, jejichž frekvence je mezi 2K a 5K. Aktivní bzučák je často dražší než pasivní, protože má v sobě zabudováno více kmitajících obvody.



Připojení

Schéma

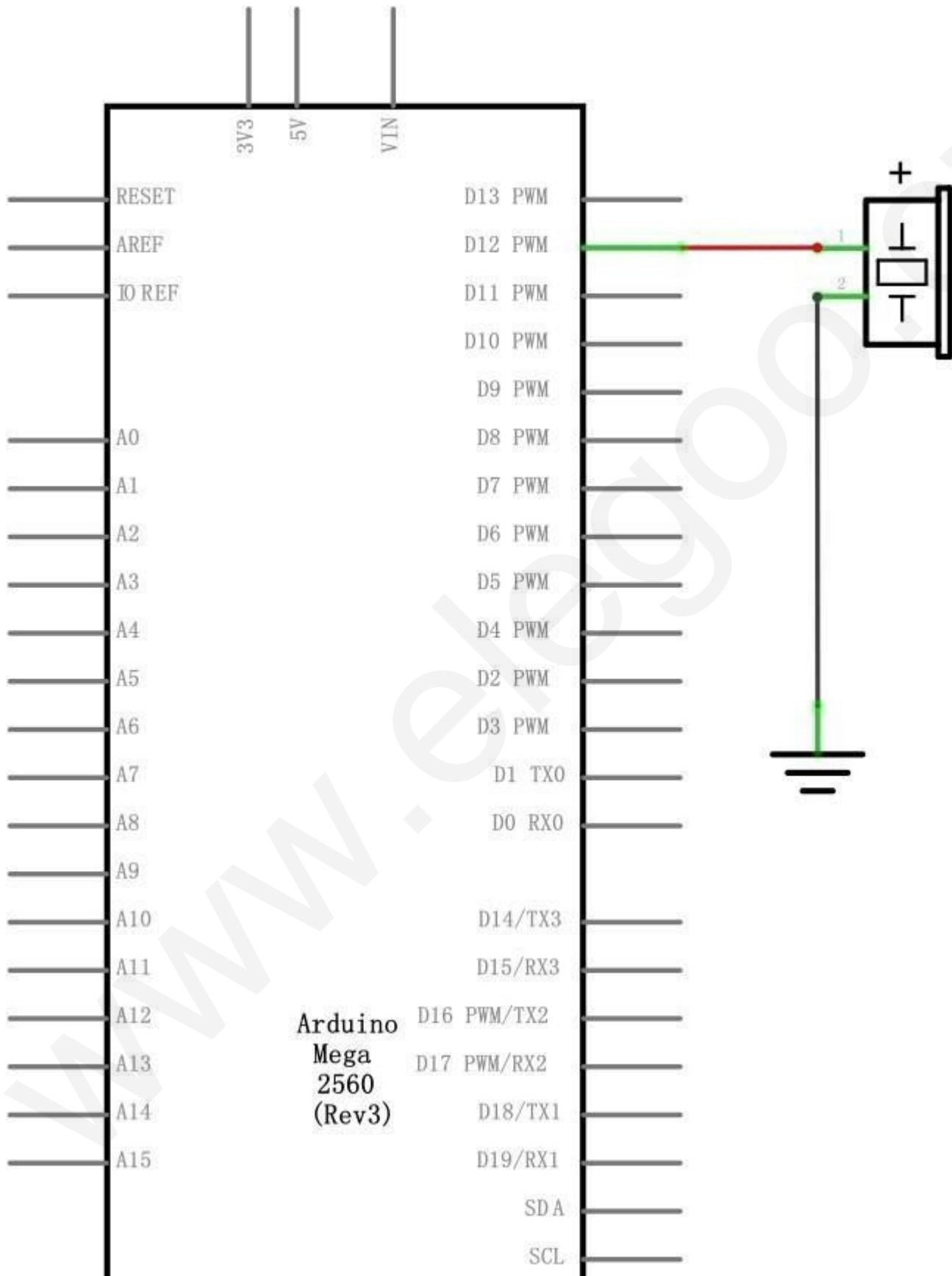
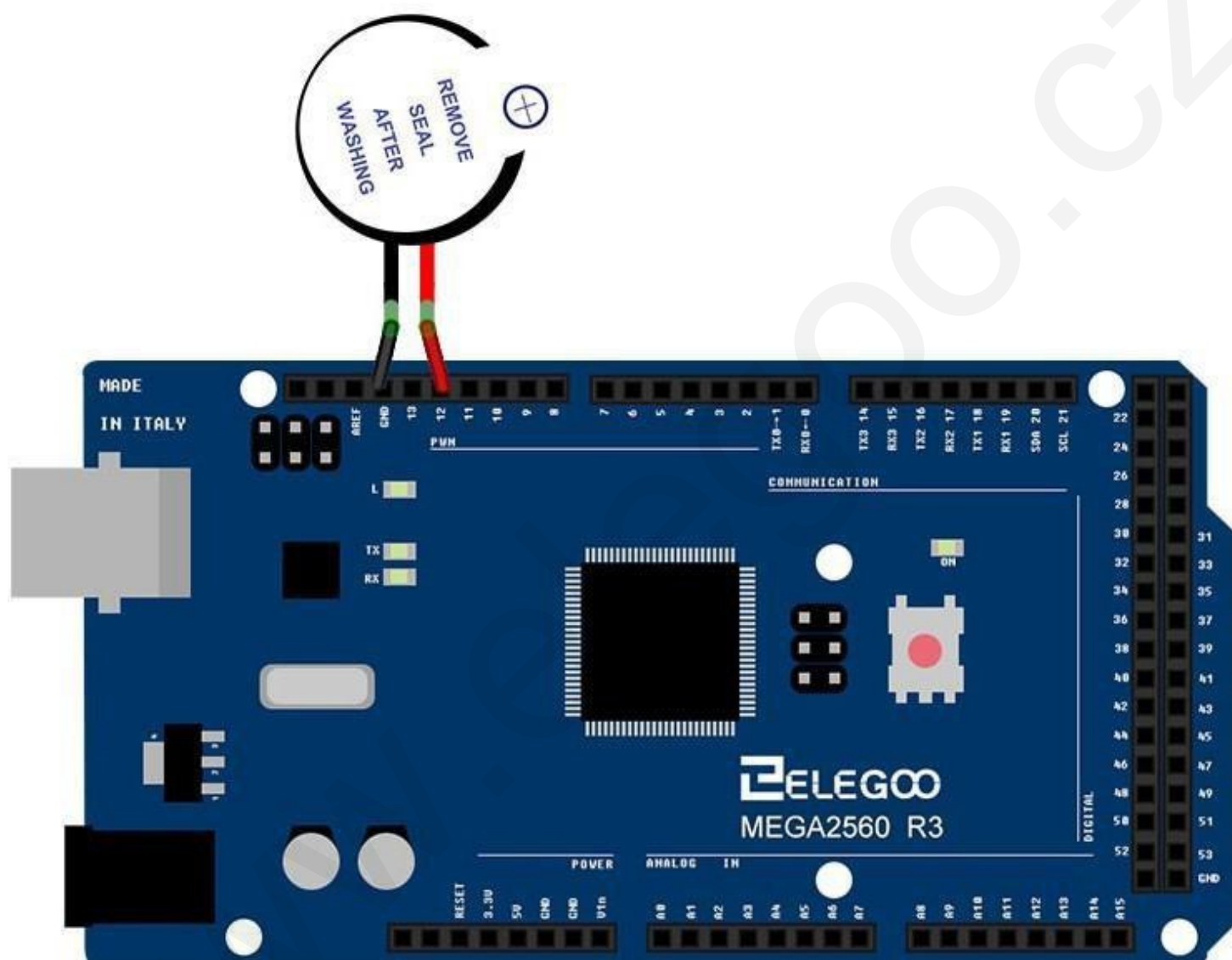


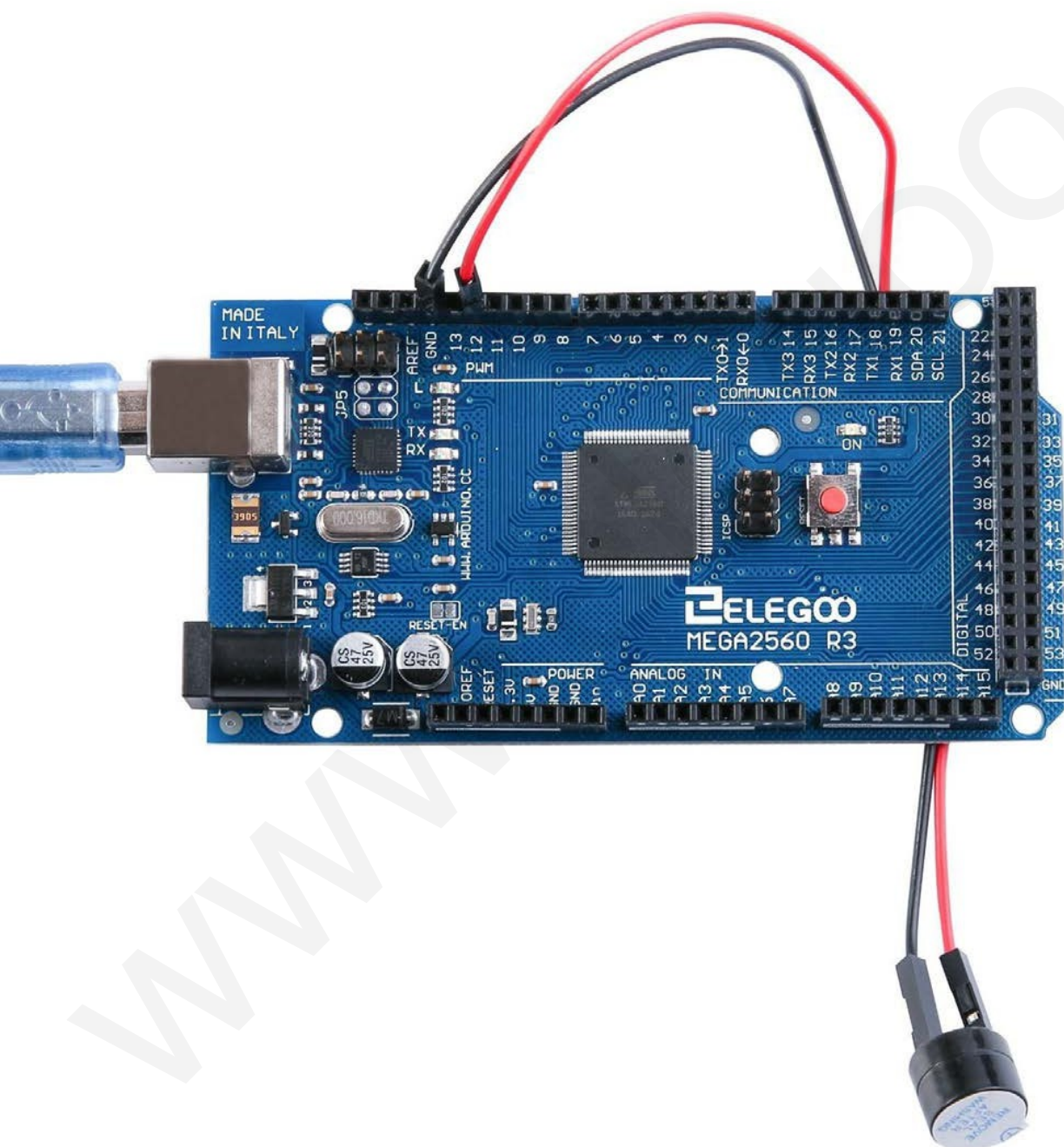
Schéma zapojení



Kód

Po zapojení otevřete program ve složce s kódem - Lekce 6 Tvorba zvuků a kliknutím na UPLOAD program nahrajte. Podrobnosti o nahrávání programu naleznete v lekci 2, pokud se vyskytnou chyby.

Příklad obrázku



Lekce 7 Pasivní bzučák

Přehled

V této lekci se naučíte používat pasivníbuzzer.

Účelem experimentu je vytvořit osm různých zvuků, přičemž každý zvuk trvá 0,5 sekundy: od altového Do (523 Hz), Re (587 Hz), Mi (659 Hz), Fa (698 Hz), So (784 Hz), La (880 Hz), Si (988 Hz) až po vysoké Do (1047 Hz).

Požadovaná součást:

- (1) x Elegoo Mega2560 R3
- (1) x pasivní bzučák
- (2) x vodiče F-M (samice na samce DuPont)

Úvod do složky

Pasivní bzučák:

Princip fungování pasivního bzučáku spočívá v tom, že pomocí PWM generuje zvuk, který rozechvívá vzduch. Vhodnou změnou frekvence vibrací může generovat různé zvuky. Například vysíláním pulzu o frekvenci 523 Hz může generovat Alto Do, pulzem o frekvenci 587 Hz může generovat střední Re, pulzem o frekvenci 659 Hz střední třídy Mi. Pomocí bzučáku můžete přehrát skladbu.

Měli bychom si dát pozor, abychom nepoužili funkci analogového zápisu () na desce MEGA 2560 R3 pro generování impulsu do bzučáku, protože výstupní impuls analogového

Write () je



může vytvářet

www.elegoo.cz

Připojení

Schéma

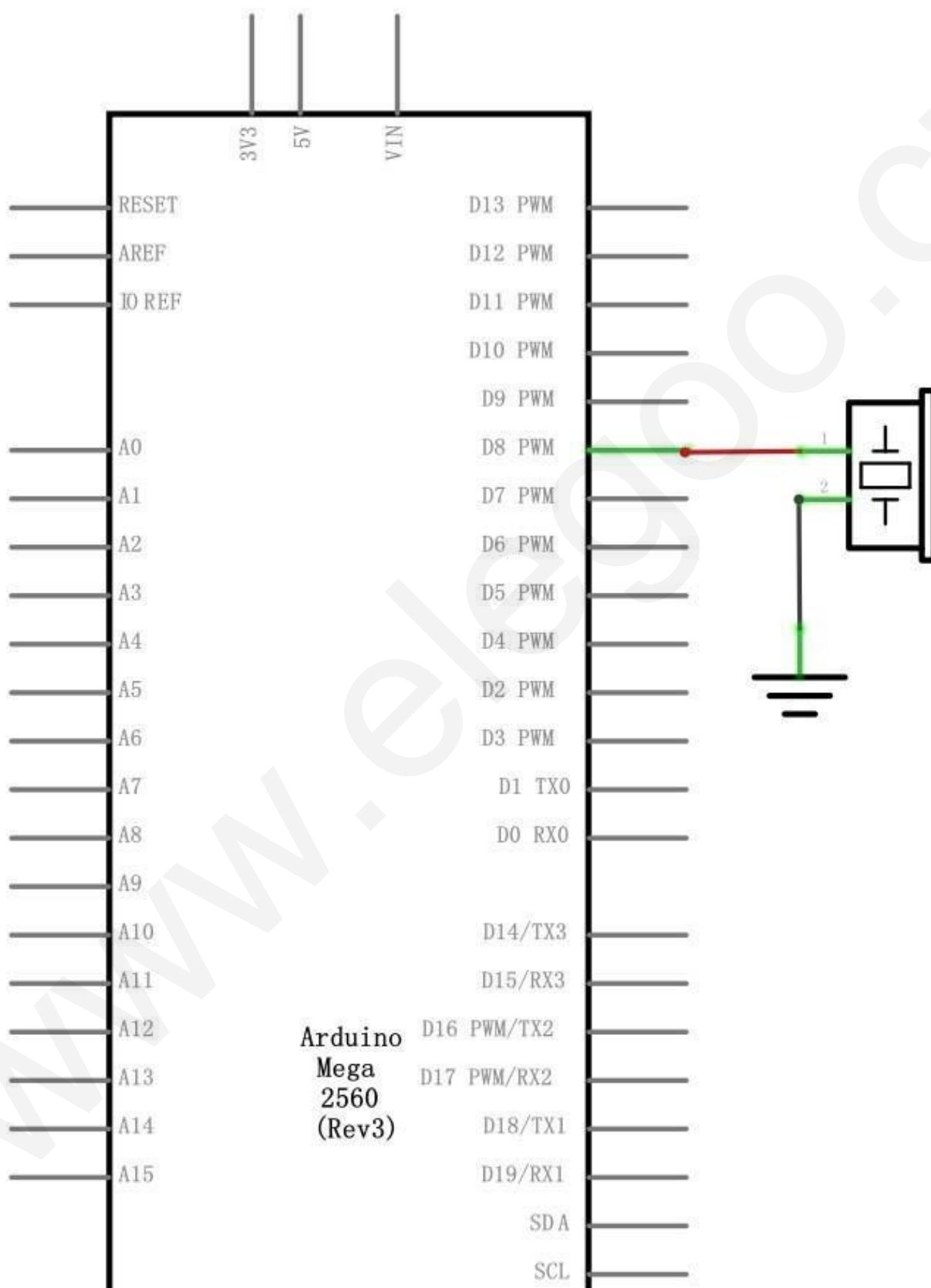
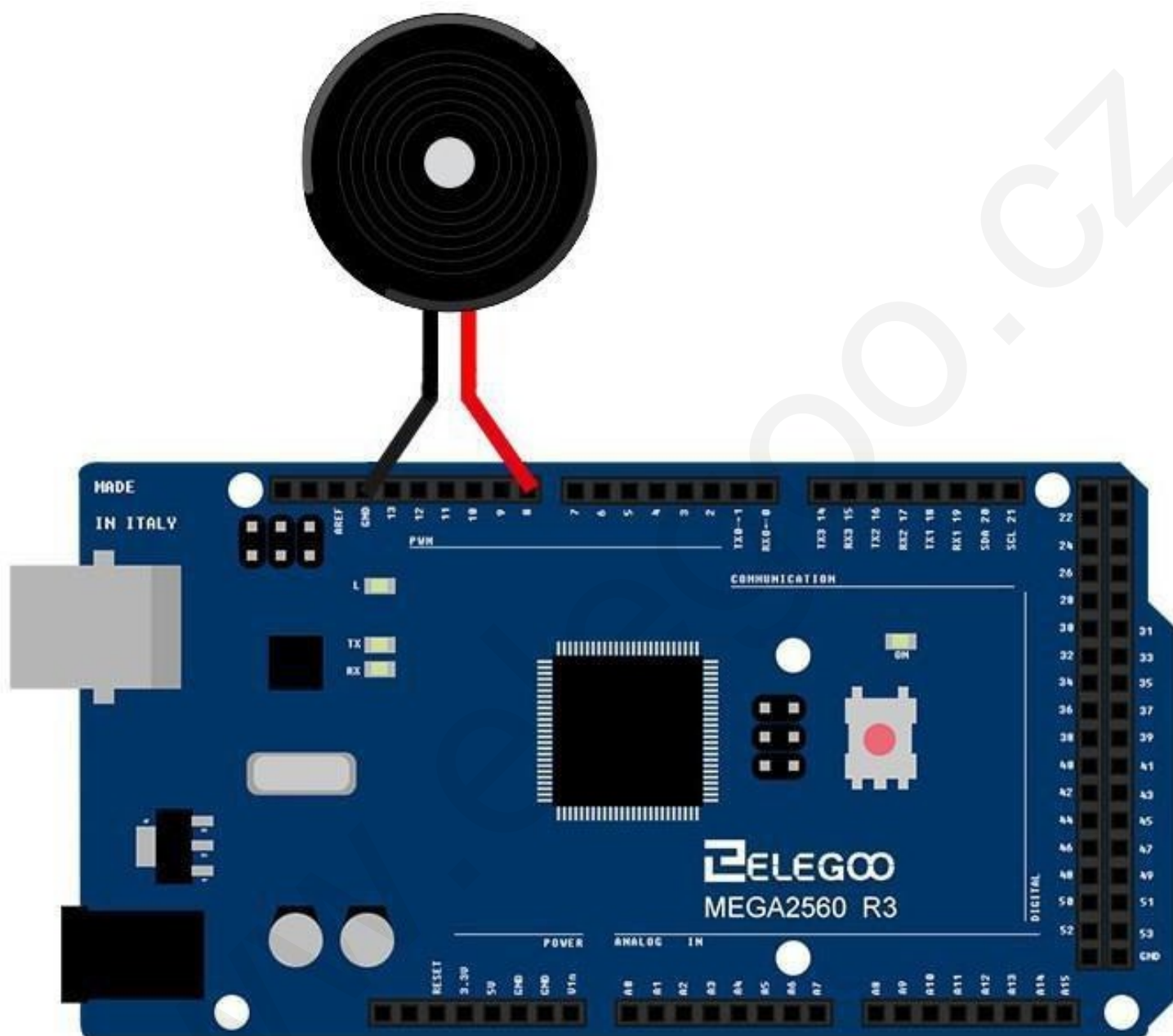


Schéma zapojení



Zapojení bzučáku připojeného k desce MEGA 2560 R3, červený vodič (kladný) na pin8, černý vodič (záporný) na GND.

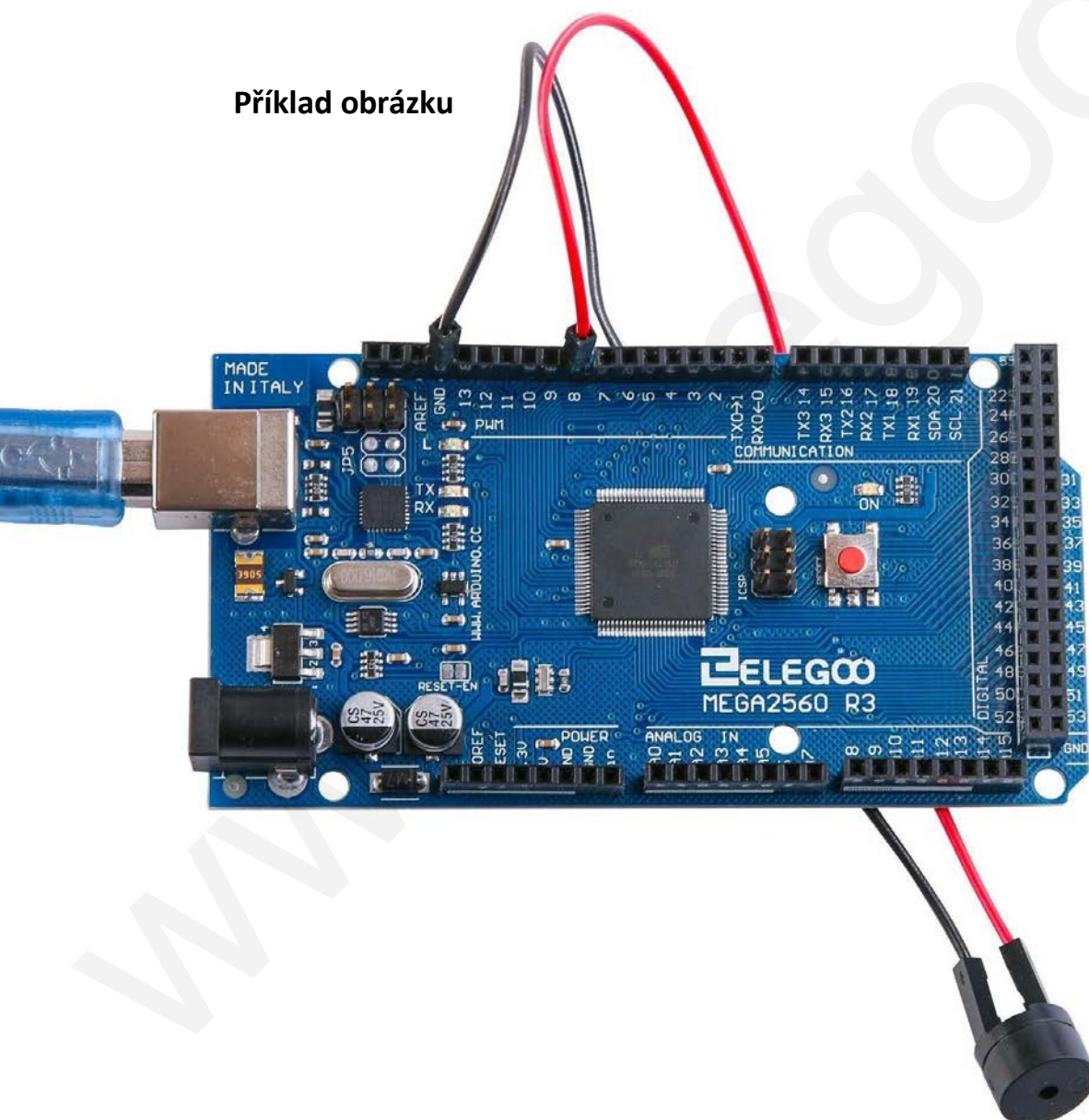
Kód

Po zapojení otevřete program ve složce s kódem - Lekce 7 Pasivní bzučák a kliknutím na UPLOAD program nahrajte. Podrobnosti o nahrávání programu naleznete v Lekci 2, pokud se vyskytnou nějaké chyby.

Před spuštěním této funkce se ujistěte, že máte nainstalovanou knihovnu <pitch>, případně ji nainstalujte znovu. Jinak váš kód nebude fungovat.

Podrobnosti o načítání souboru knihovny naleznete v lekci 1.

Příklad obrázku



Lekce 8 Přepínač naklápěcí koule

Přehled

V této lekci se dozvíte, jak používat náklonový kuličkový spínač k detekci malého úhlu sklonu.

Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x přepínač naklápěcí koule
- (2) x vodiče F-M (samice na samce DuPont)



Úvod do složky

Snímač náklonu:

Senzory náklonu (kulový spínač náklonu) umožňují zjistit orientaci nebo náklon. Jsou malé, levné, mají nízkou spotřebu energie a snadno se používají. Při správném používání se neopotřebovávají. Díky své jednoduchosti jsou oblíbené pro hračky, pomůcky a spotřebiče. Někdy se ze zřejmých důvodů označují jako "rtuťové spínače", "spínače náklonu" nebo "senzory valivé koule".

Obvykle jsou tvořeny nějakou dutinou (oblíbený je válcový tvar, i když ne vždy) s volnou vodivou hmotou uvnitř, například kapkou rtuti nebo valivou koulí. Na jednom konci dutiny jsou dva vodivé prvky (póly). Když je snímač orientován tak, že je tento konec směrem dolů, hmota se na póly naválí a zkratuje je, což funguje jako spínací tlačítko.

Přestože nejsou tak přesné a flexibilní jako plnohodnotný akcelerometr, mohou náklonové spínače detekovat pohyb nebo orientaci. Další výhodou je, že ty velké mohou samy přepínat napájení. Akcelerometry naproti tomu vysílají digitální nebo analogové napětí, které se pak musí analyzovat pomocí dodatečných obvodů.

Připojení

Schéma

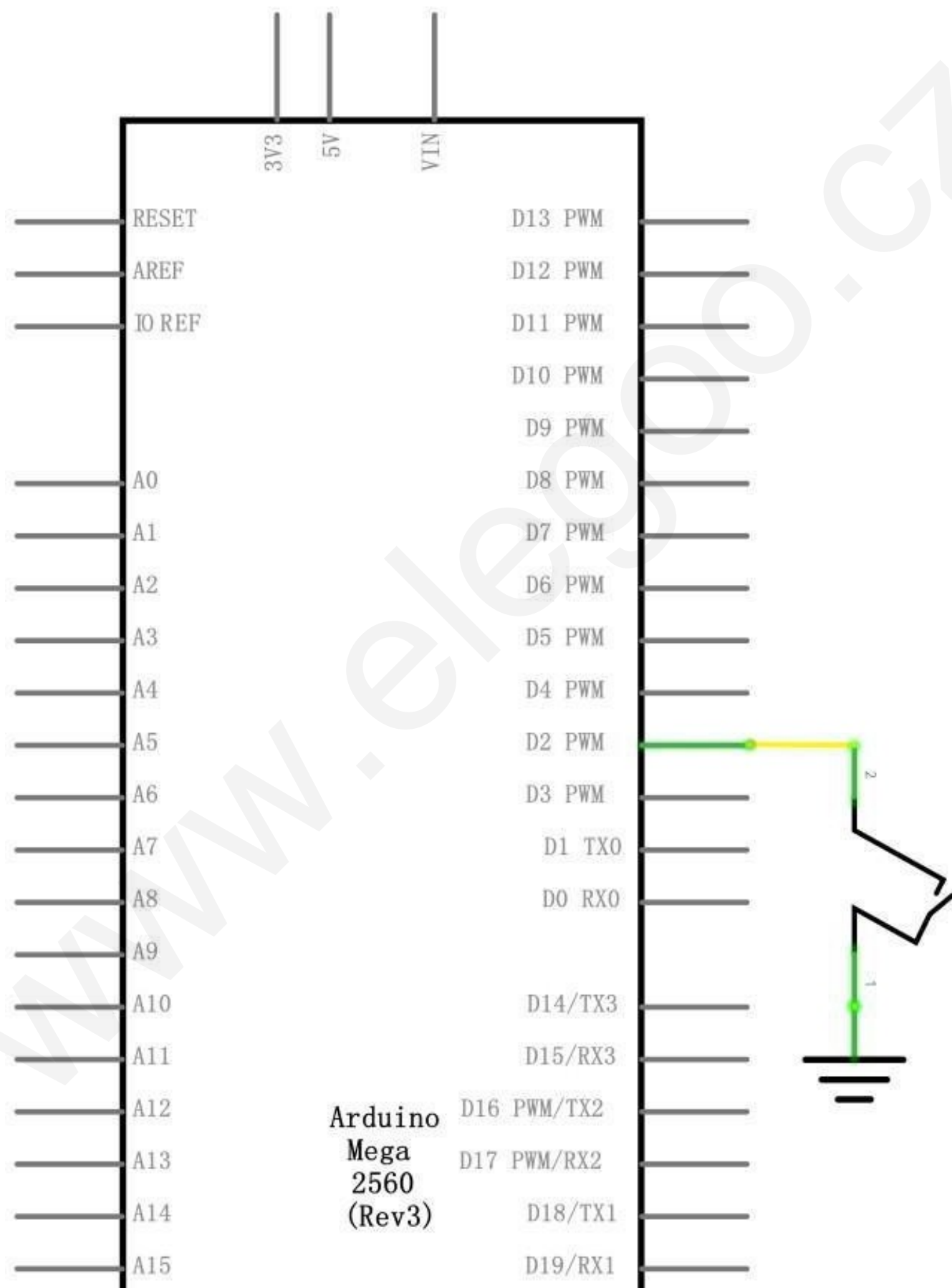
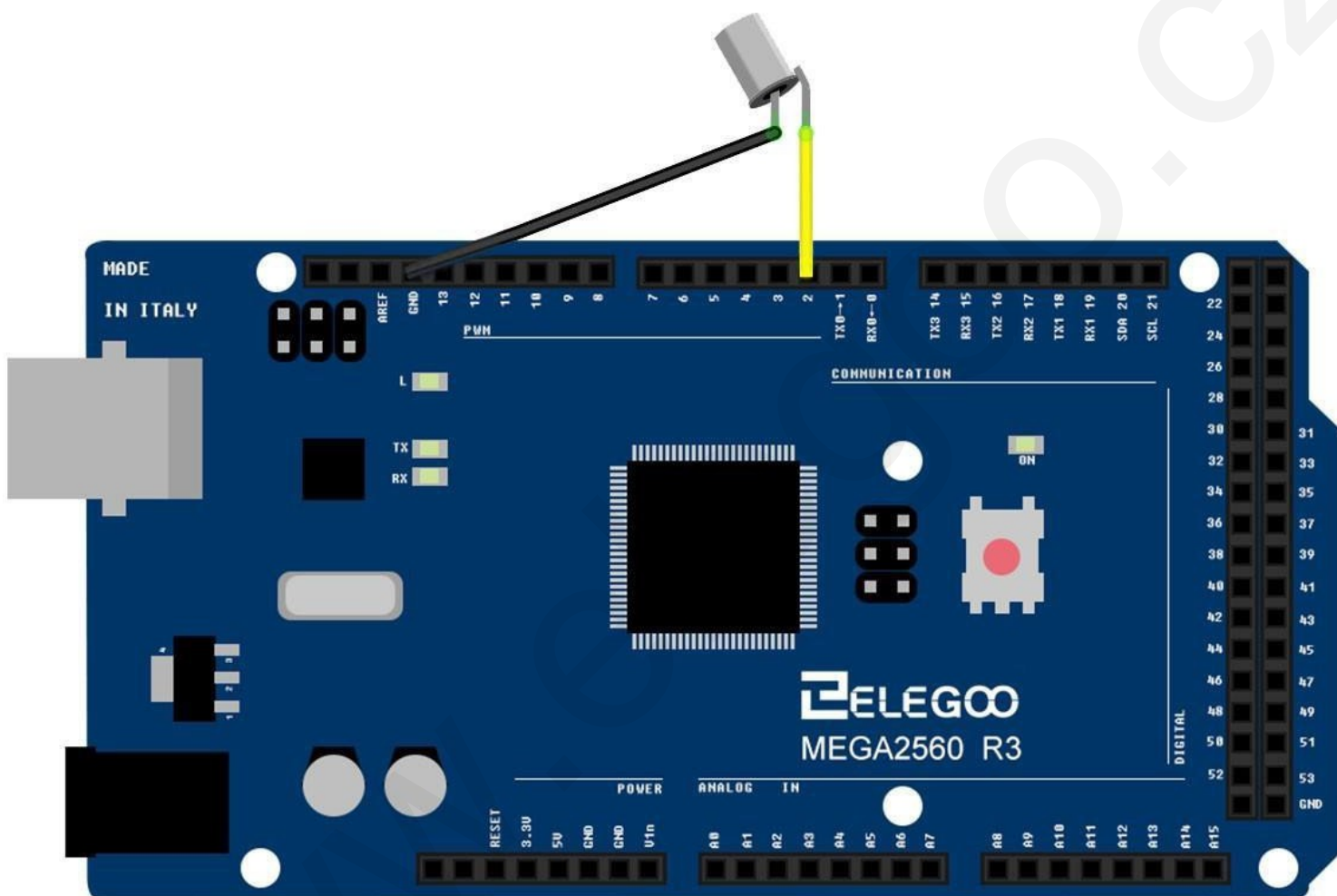


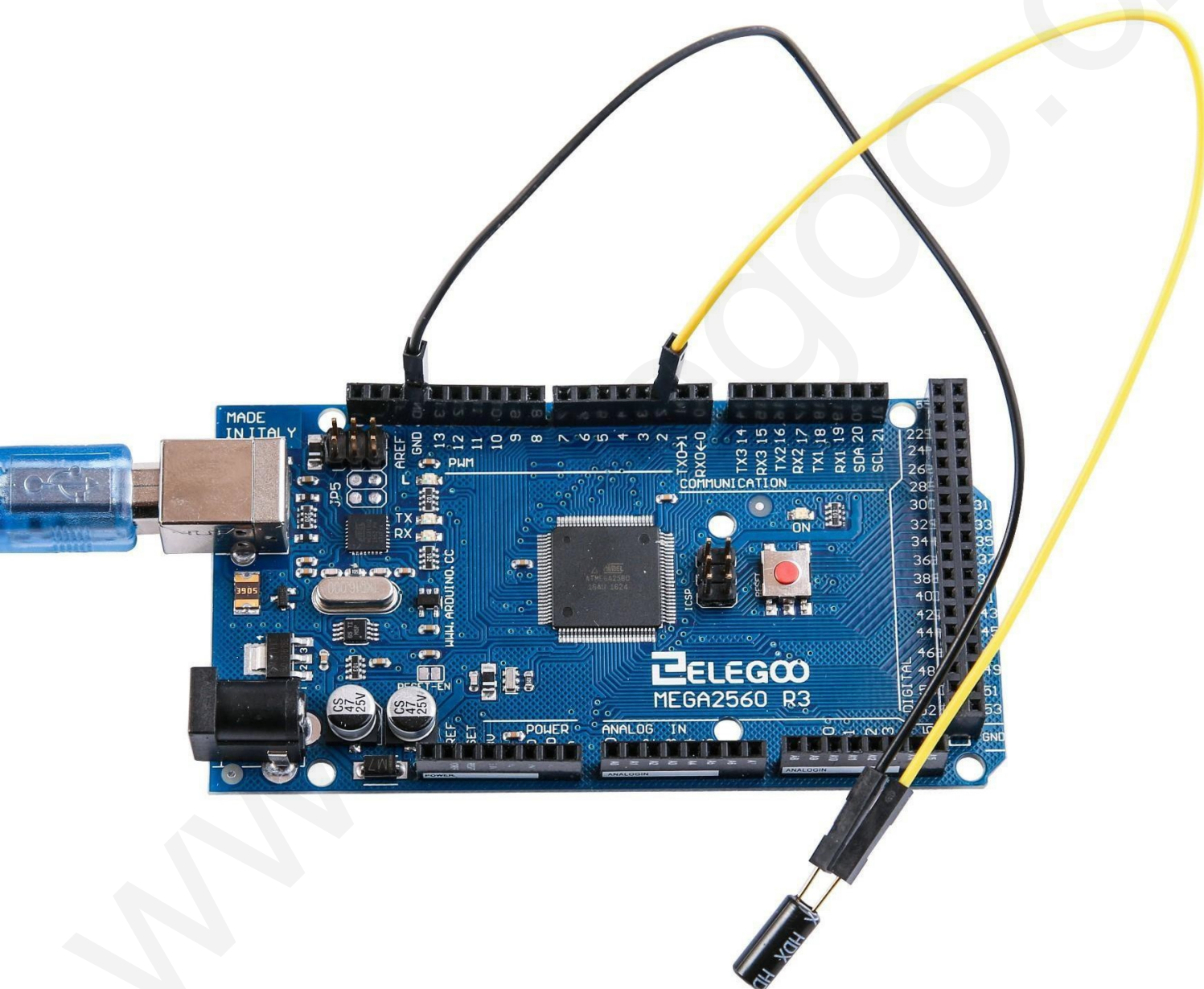
Schéma zapojení



Kód

Po zapojení otevřete program ve složce s kódem - Lesson 8 Ball Switch a kliknutím na UPLOAD program nahrajte. Podrobnosti o nahrávání programu naleznete v lekci 2, pokud se vyskytnou nějaké chyby.

Příklad obrázku



Lekce 9 LCD displej

Přehled

V této lekci se naučíte zapojit a používat alfanumerický LCD displej. Displej má podsvícení LED a může zobrazovat dva řádky s až 16 znaky na každém řádku. Na displeji vidíte obdélníky pro jednotlivé znaky a pixely, které jednotlivé znaky tvoří.

Displej je pouze bílý na modrém podkladu a je určen k zobrazování textu.

V této lekci spustíme ukázkový program Arduina pro knihovnu LCD, ale v příští lekci pomocí senzorů přimějeme náš displej zobrazovat teplotu.

Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x modul LCD1602
- (1) x potenciometr (10k)
- (1) x 830 vázacích bodů Deska na chleba
- (16) x vodiče M-M (propojovací vodiče typu Male to Male)



Úvod do složky

LCD1602

Seznámení s vývody LCD1602:

VSS: Vývod, který se připojuje k zemi

VDD: Vývod, který se připojuje k napájecímu zdroji +5 V.

VO: Vývod pro nastavení kontrastu LCD1602

RS: Vývod pro výběr registru, který řídí, do které části paměti LCD se zapisují data. Můžete zvolit buď datový registr, do kterého se zapisuje to, co se zobrazí na obrazovce, nebo instrukční registr, ve kterém řadič LCD hledá instrukce, co má dále dělat.

R/W: Vývod pro čtení/zápis, který volí režim čtení nebo zápisu.

E: Povolovací pin, který po dodání nízkourovňové energie způsobí, že modul LDC provede příslušné instrukce.

D0-D7: Piny pro čtení a zápis dat

A a K: Piny pro ovládání podsvícení LED diodami

Připojení Schéma

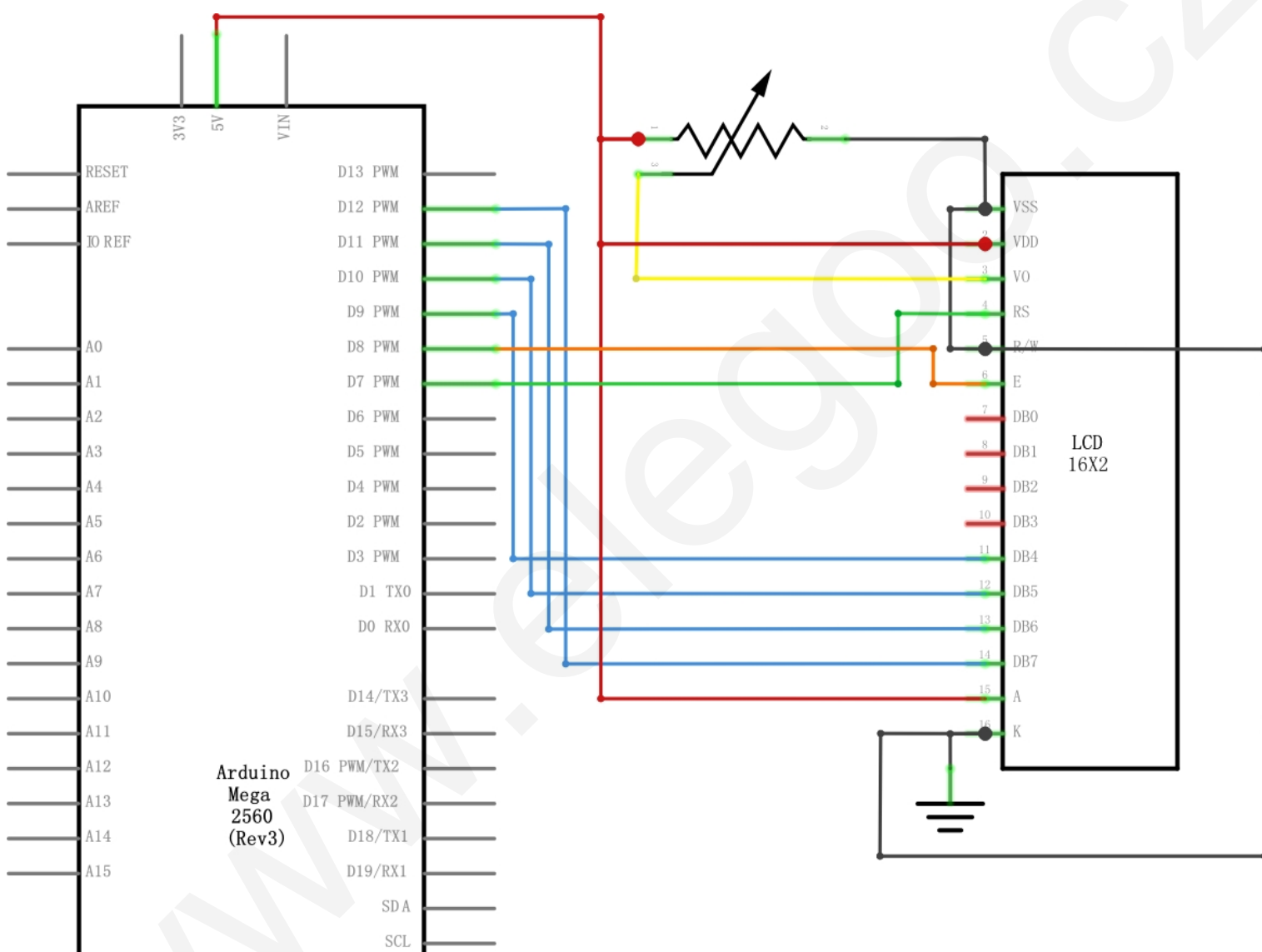
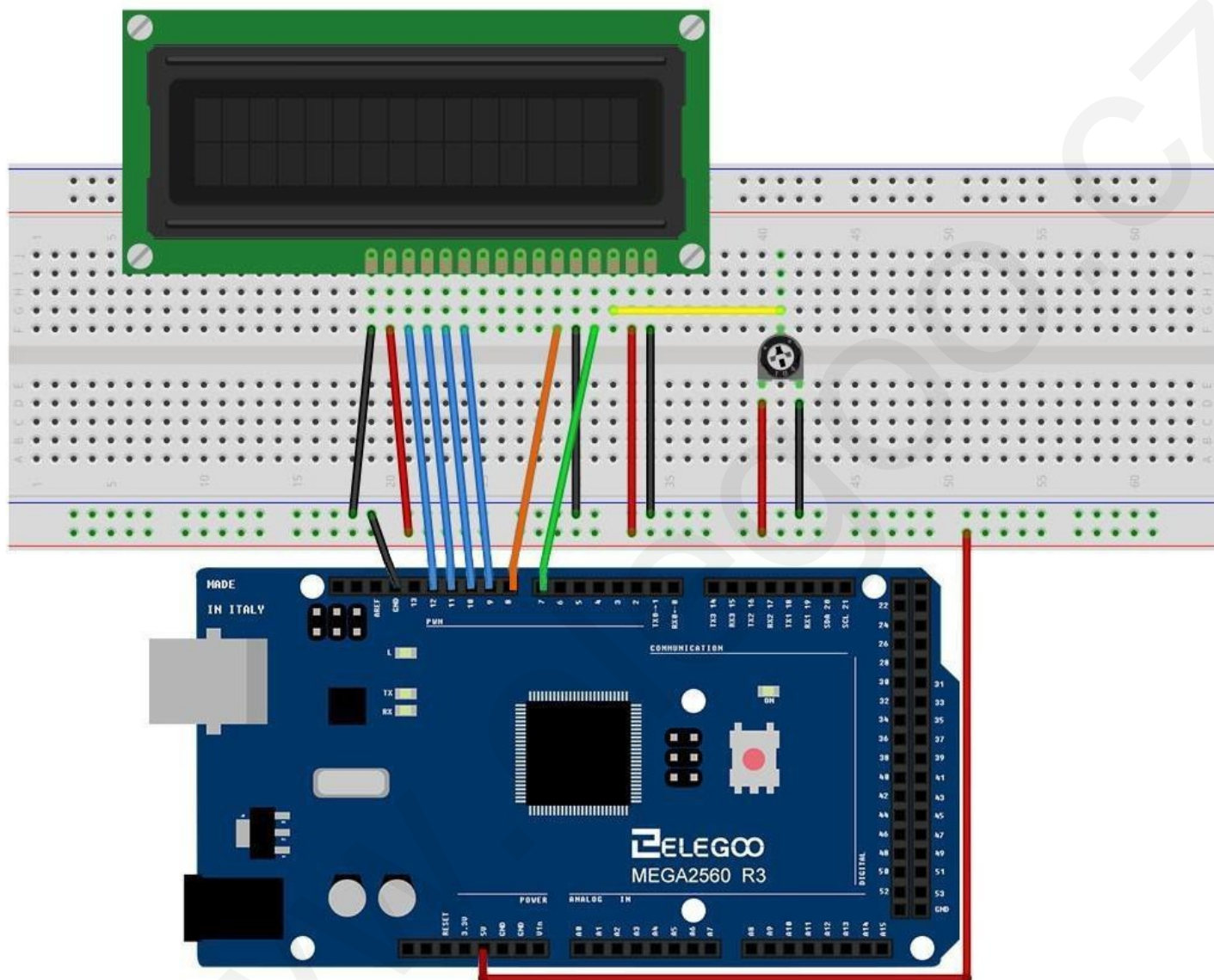


Schéma zapojení



LCD displej potřebuje šest pinů Arduina, které jsou nastaveny jako digitální výstupy. Potřebuje také připojení 5V a GND.

Existuje řada souvislostí. Zarovnání displeje s horní částí desky pomáhá identifikovat jeho piny bez dlouhého počítání, zejména pokud má deska očíslované řady, přičemž řada 1 je horní řadou desky. Nezapomeňte na dlouhý žlutý vodič, který spojuje posuvník potenciometru s pinem 3 displeje. Hrnc" slouží k ovládání kontrastu displeje.

Může se stát, že váš displej je dodáván bez připojených záhlavních pinů. V takovém případě postupujte podle pokynů v následující části.

Kód

Po zapojení otevřete program ve složce s kódem - Lekce 9 LCD displej a kliknutím na UPLOAD program nahrajte. Podrobnosti o nahrávání programu naleznete v Lekci 2, pokud se vyskytnou chyby.

Před spuštěním této funkce se ujistěte, že máte nainstalovanou knihovnu < Liquid Crystal >, případně ji nainstalujte znovu. Jinak váš kód nebude fungovat.

Podrobnosti o načítání souboru knihovny naleznete v lekci 1.

Nahrajte kód na desku Arduino a měla by se zobrazit zpráva "hello, world", následovaná číslem, které se počítá od nuly.

Na náčrtu je nejprve třeba upozornit na čáru:

```
#include <LiquidCrystal.h>
```

Tím sdělíte Arduinu, že si přejeme použít knihovnu Liquid Crystal.

Dále máme řádek, který jsme museli upravit. Ten definuje, které piny Arduina mají být připojeny ke kterým pinům displeje.

```
LiquidCrystal lcd(7, 8, 9, 10, 11,12);
```

Po nahrání tohoto kódu se ujistěte, že podsvícení svítí, a nastavte potenciometr po celé délce, dokud se nezobrazí textová zpráva.

Ve funkci 'setup' máme dva příkazy:

```
lcd.begin(16, 2);
```

```
lcd.print("Hello,World!");
```

První z nich říká knihovně tekutých krystalů, kolik má displej sloupců a řádků.

Na druhém řádku se zobrazí zpráva, kterou vidíme na prvním řádku obrazovky.

Ve funkci 'loop' máme také dva příkazy:

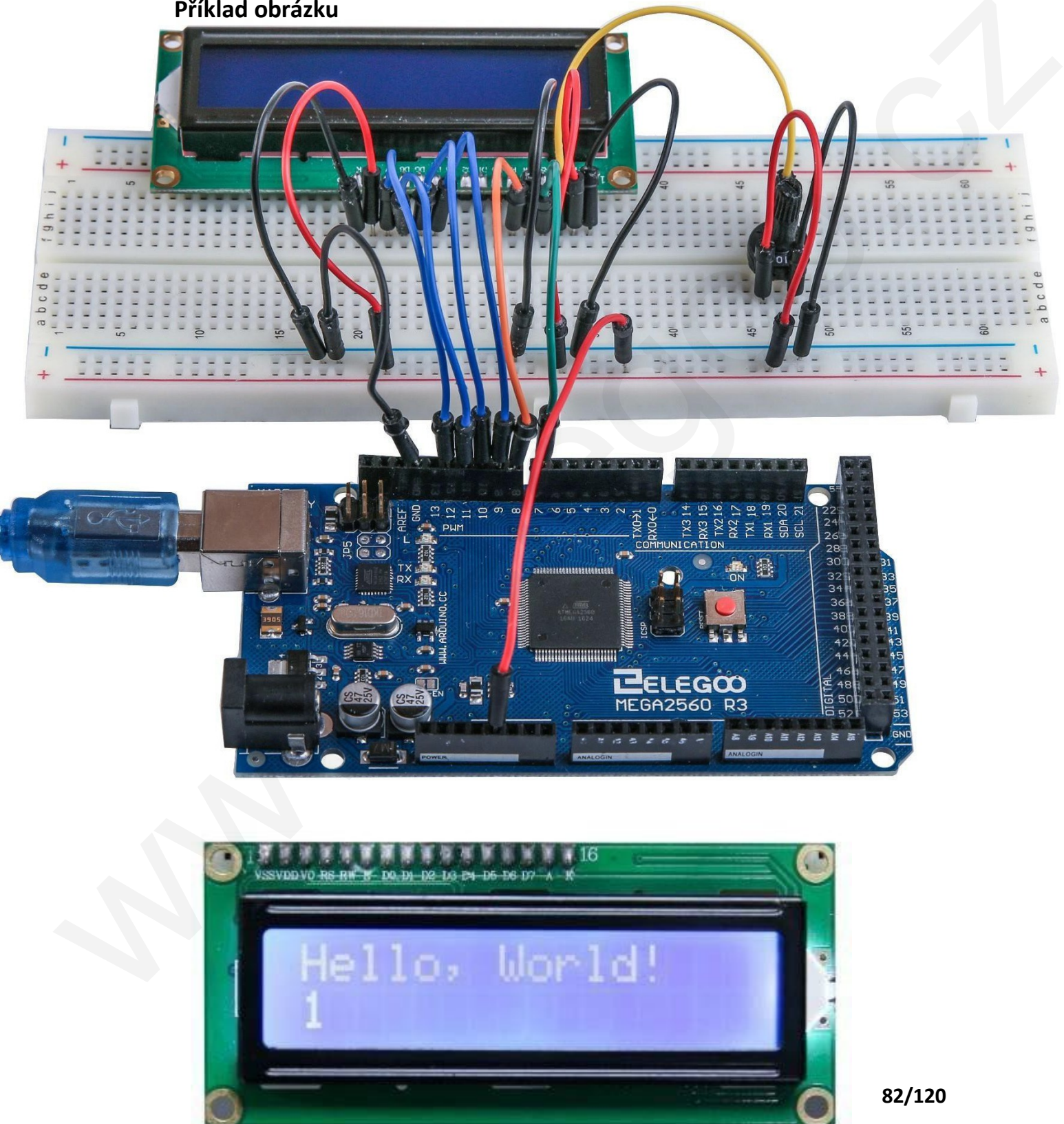
```
lcd.setCursor(0, 1);
```

```
lcd.print(millis()/1000);
```

První nastaví pozici kurzoru (kde se zobrazí další text) na sloupec 0 a řádek 1. Čísla sloupců i řádků začínají od 0, nikoli od 1.

Druhý řádek zobrazuje počet milisekund od resetování Arduina.

Příklad obrázku



Lekce 10 Teploměr

Přehled

V této lekci budete používat LCD displej k zobrazení teploty.

Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x modul LCD1602
- (1) x 10k ohmový rezistor
- (1) x termistor
- (1) x potenciometr
- (1) x 830 vázacích bodů Deska na chleba
- (18) x vodiče M-M (propojovací vodiče)

Úvod do komponent

Termistor

Termistor je tepelný rezistor - rezistor, který mění svůj odpor s teplotou. Technicky vzato jsou všechny rezistory termistory - jejich odpor se mírně mění s teplotou - ale tato změna je obvykle velmi malá a obtížně měřitelná. Termistory jsou vyrobeny tak, že se jejich odpor s teplotou mění drasticky, takže může jít o změnu 100 ohmů nebo více na jeden stupeň!

Existují dva druhy termistorů, NTC (záporný teplotní koeficient) a PTC (kladný teplotní koeficient). Obecně se k měření teploty používají snímače NTC. PTC se často používají jako resetovatelné pojistky - zvýšením teploty se zvýší jejich odpor, což znamená, že se při průchodu většího proudu zahřejí a "přiškrtní" proud, což je docela užitečné pro ochranu obvodů.

Připojení Schéma

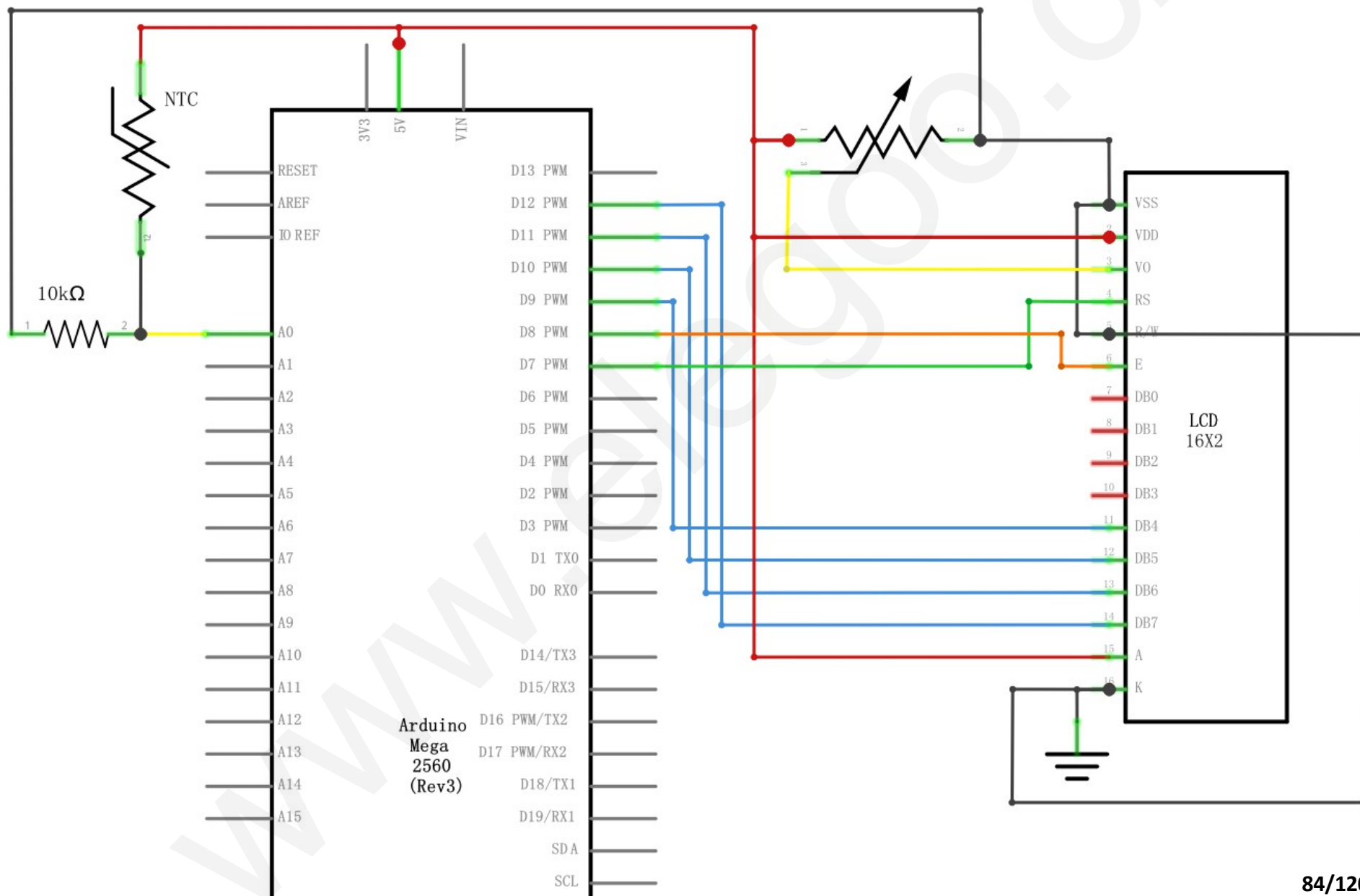
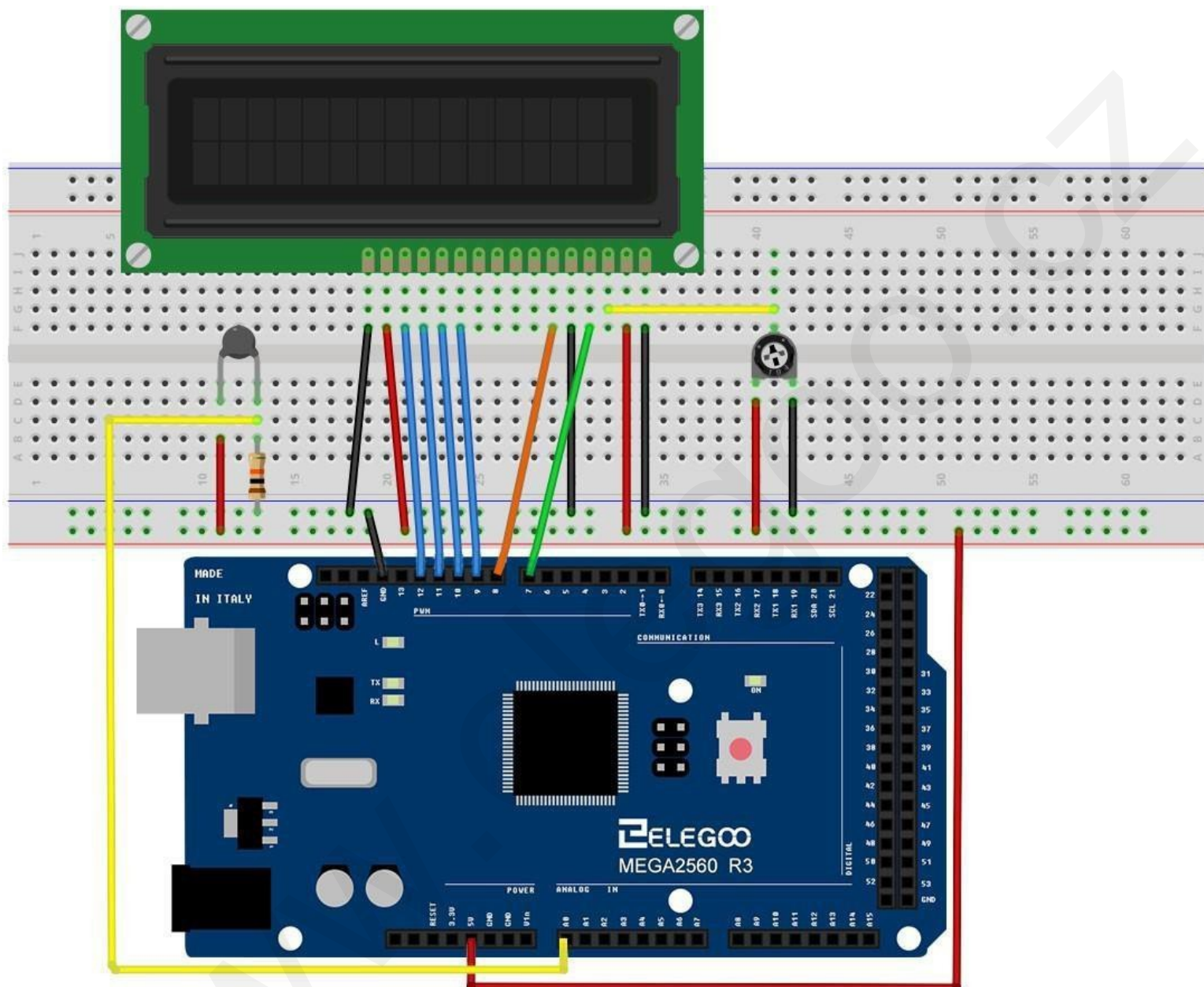


Schéma zapojení



Rozložení na desce vychází z rozvržení z lekce 22, takže se zjednoduší, pokud ho máte stále na desce.

V blízkosti potenciometru je několik propojovacích vodičů, které byly v tomto uspořádání mírně posunuty.

Rezistor 10 kΩ a termistor jsou na desce nové.

Kód

Po zapojení otevřete program ve složce s kódem - Lekce 10 Teploměr a kliknutím na UPLOAD program nahrajte. Podrobnosti o nahrávání programu naleznete v Lekci 2, pokud se vyskytnou chyby.

Před spuštěním se ujistěte, že máte nainstalovanou knihovnu < LiquidCrystal >, případně ji nainstalujte znovu. Jinak váš kód nebude fungovat.

Podrobnosti o načítání souboru knihovny naleznete v lekci 1.

Náčrtek pro tuto lekci vychází z lekce 22. Nahrajte ji do Arduina a měli byste zjistit, že zahřátí teplotního čidla přiložením prstu na něj zvýší hodnotu teploty.

Považuji za užitečné vložit nad příkaz 'lcd' řádek s komentářem.

```
// BS E D4 D5 D6 D7
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11,12);
```

To vám usnadní práci, pokud se rozhodnete změnit používané kolíky.

Ve funkci 'loop' se nyní dějí dvě zajímavé věci. Zaprvé musíme převést analogové údaje ze snímače teploty na skutečnou teplotu a zadruhé musíme vymyslet, jak je zobrazit.

Nejprve se podívejme na výpočet teploty. `int`

```
tempReading =analogRead(tempPin);
```

```
double tempK = log(10000.0 * ((1024.0 / tempReading -1)));
```

```
tempK = 1 / (0,001129148 + (0,000234125 + (0,0000000876741 * tempK * tempK ))  
* tempK );
```

```
float tempC = tempK - 273,15;
```

```
float tempF = (tempC * 9,0)/ 5,0 +32,0;
```

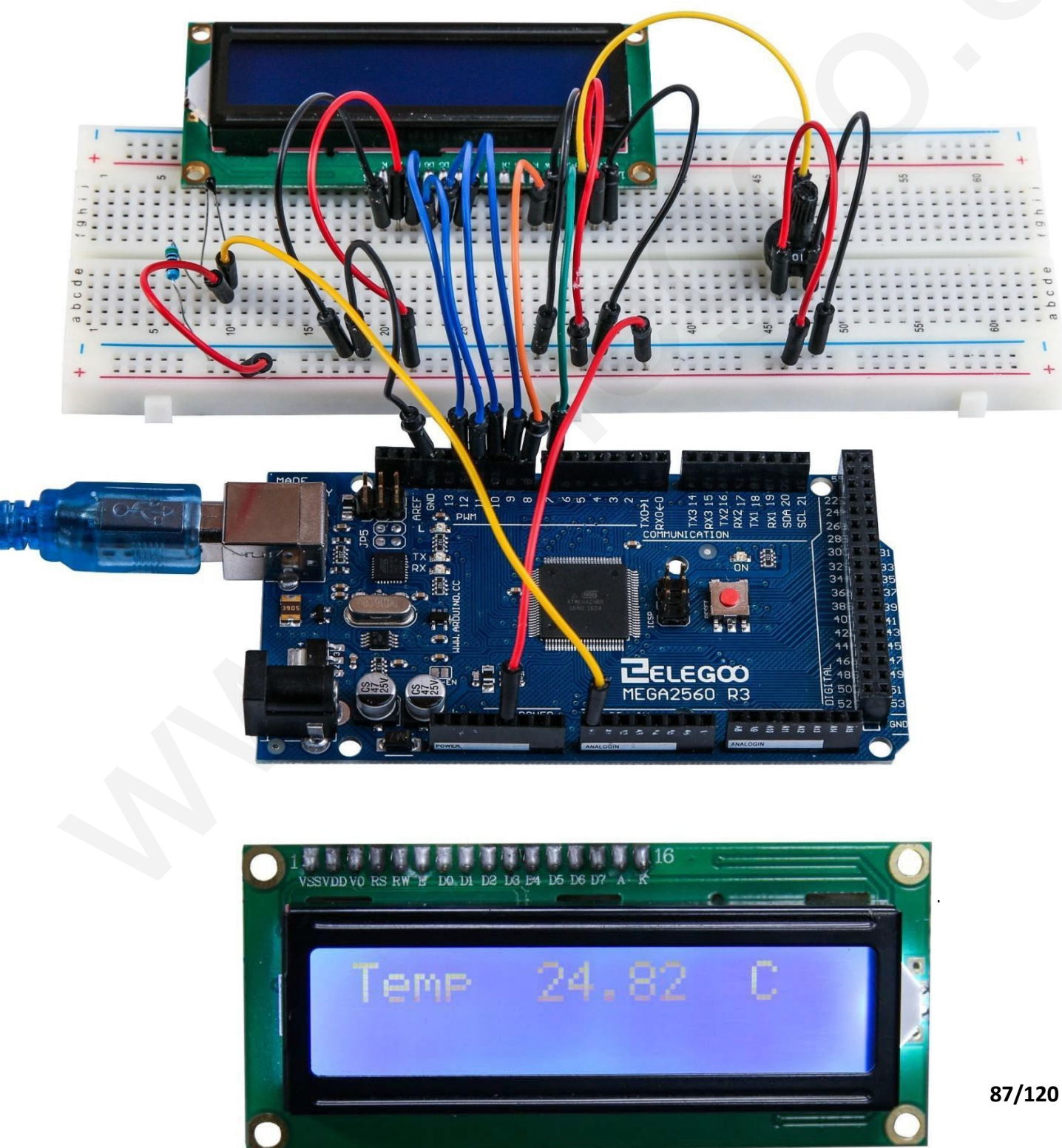
Zobrazení měnících se hodnot na displeji LCD může být složité. Hlavním problémem je, že údaj nemusí mít vždy stejný počet číslic. Pokud se tedy teplota změnila ze 101,50 na 99,00, hrozí, že na displeji zůstane číslice navíc ze starého údaje.

Abyste tomu předešli, vypište pokaždé celý řádek LCD kolem smyčky.

```
lcd.setCursor(0, 0);  
lcd.print("Temp      C  ");  
lcd.setCursor(6, 0);  
lcd.print(tempF);
```

Poněkud zvláštní komentář slouží k připomenutí 16 sloupců displeje. V místě, kde se bude odečítat skutečný údaj, pak můžete vypsat řetězec této délky s mezerami. Chcete-li vyplnit prázdná místa, nastavte pozici kurzoru na místo, kde se má údaj zobrazit, a poté jej vytiskněte.

Příklad obrázku



Lekce 11 Osm LED s 74HC595

Přehled

V této lekci se dozvíte, jak používat osm velkých červených LED s MEGA2560, aniž byste se museli vzdát 8 výstupních pinů!

Ačkoli byste mohli připojit osm LED diod, každou s rezistorem, k jednomu pinu MEGA2560, rychle by vám začaly docházet piny na MEGA2560. Pokud nemáte k MEGA2560 připojeno mnoho věcí. Je to v pořádku - ale často se stává, že chceme mít tlačítka, senzory, serva atd. a než se nadějete, nezbydou vám žádné piny. Místo toho tedy použijete čip 74HC595, který se nazývá převodník sériového na paralelní rozhraní. Tento čip má osm výstupů (perfektní) a tři vstupy, kterými do něj přivádíte data po částech.

Díky tomuto čipu je ovládání LED diod o něco pomalejší (LED diody můžete měnit jen asi 500 000krát za sekundu místo 8 000 000 za sekundu), ale stále je to opravdu rychlé, mnohem rychlejší, než je člověk schopen detekovat, takže to stojí za to!

Požadovaná součást:

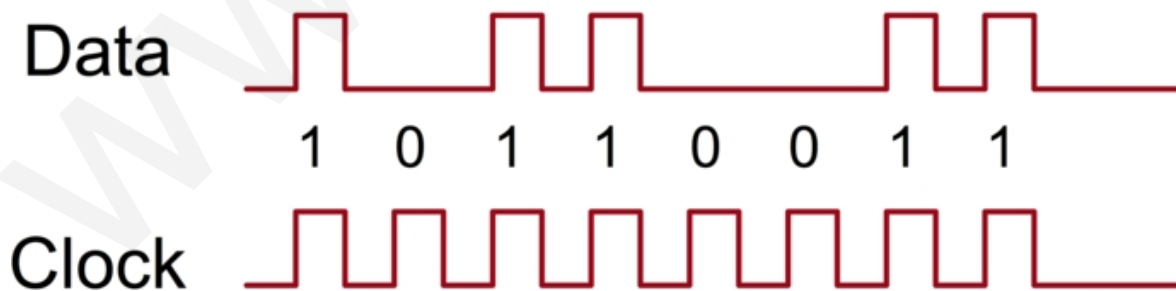
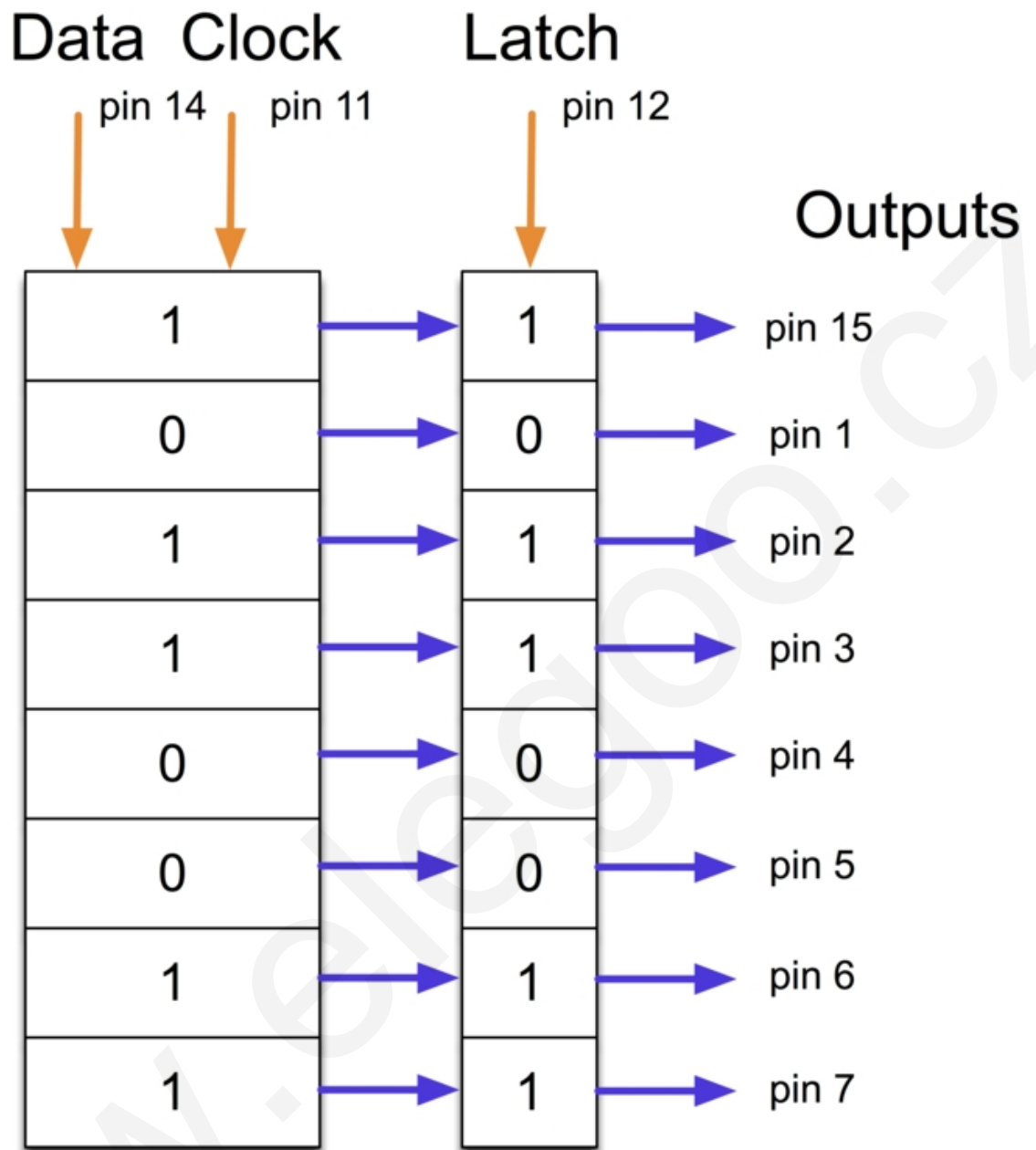
- (1) x Elegoo Mega 2560 R3
- (1) x 830 vázacích bodů deska na chleba
- (8) x LED diody
- (8) x 220 ohmů rezistorů
- (1) x 74hc595 IC
- (14) x vodiče M-M (propojovací vodiče typu Male to Male)



Úvod do složky

74HC595 Shift Register:

Posunovací registr je typ čipu, který obsahuje osm paměťových míst, z nichž každé může mít hodnotu 1 nebo 0. Pro zapnutí nebo vypnutí každé z těchto hodnot přivádíme data pomocí pinů "Data" a "Clock" čipu.



Na hodinový pin musí přijít osm impulsů. Pokud je datový pin vysoký, při každém impulsu se do posuvného registru vloží 1, jinak 0. Po přijetí všech osmi impulsů se zapnutím kolíku "Latch" zkopíruje těchto osm hodnot do registru latch. To je nezbytné; jinak by při načítání dat do posuvného registru blikaly nesprávné LED diody.

Čip má také pin pro povolení výstupů (OE), který slouží k povolení nebo zakázání všech výstupů najednou. Můžete jej připojit k vývodu MEGA 2560 podporujícímu PWM a pomocí "analogového zápisu" ovládat jas LED diod. Tento pin je aktivní, takže ho připojíme k GND.

Připojení

Schéma

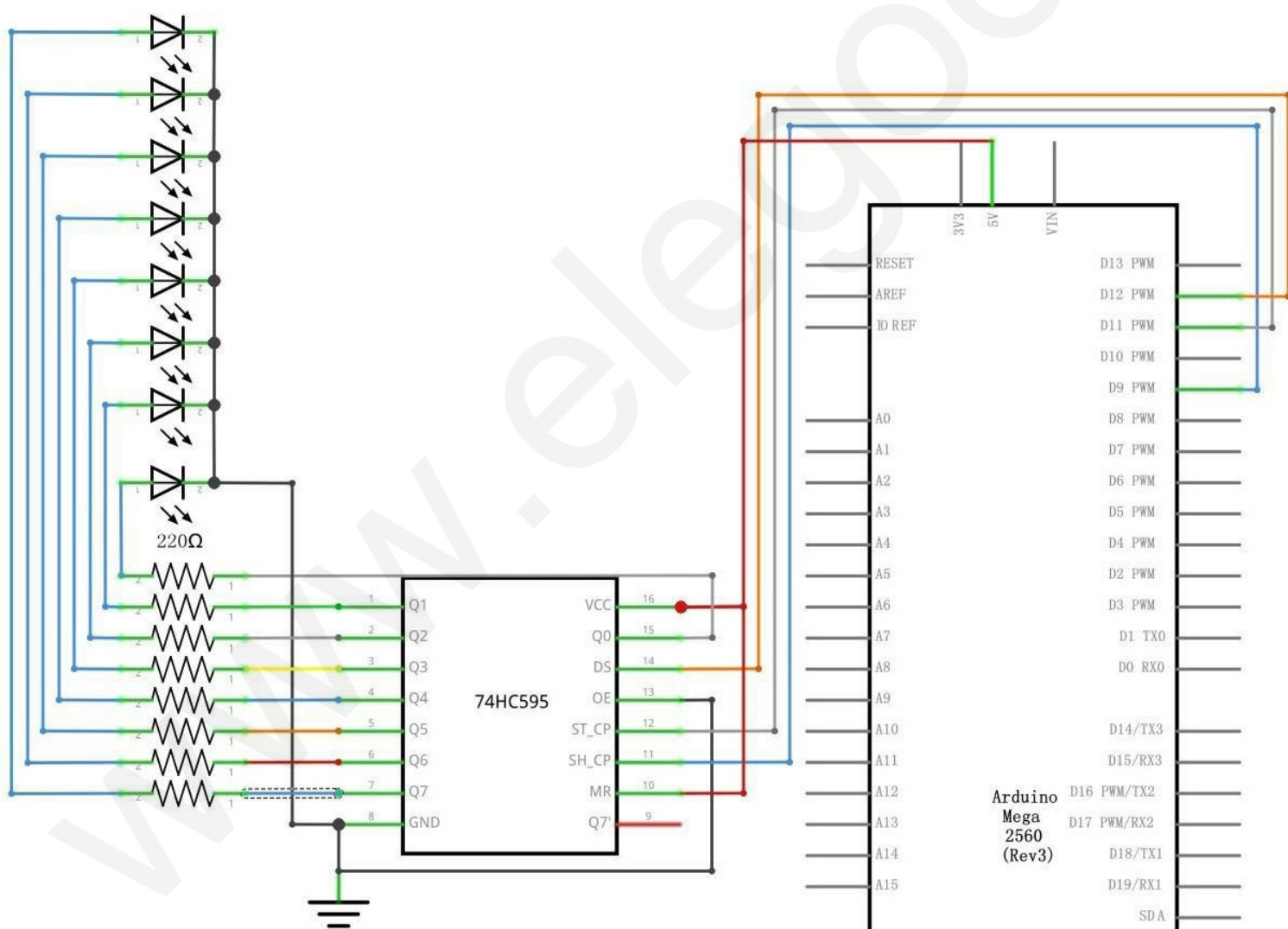
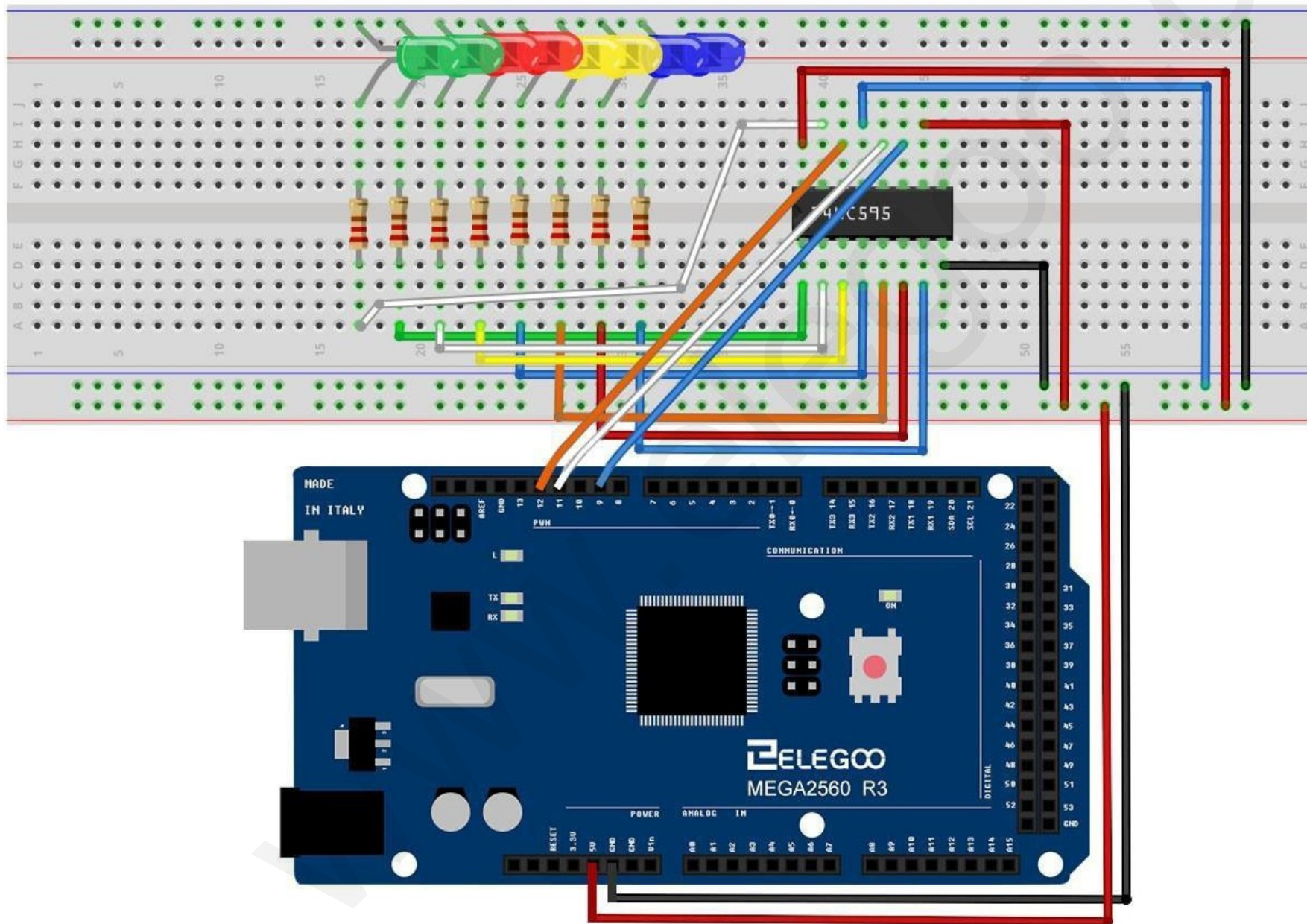


Schéma zapojení



Vzhledem k tomu, že máme osm LED diod a osm rezistorů, které je třeba připojit, je třeba provést poměrně hodně spojů.

Pravděpodobně nejjednodušší je nejprve vložit čip 74HC595, protože se k němu připojuje téměř vše ostatní. Umístěte jej tak, aby malý výřez ve tvaru písmene U směřoval k horní části desky. Pin 1 čipu je vlevo od tohoto zářezu.

Digitál 12 z MEGA2560 jde na pin #14 posuvného registru Digitál 11 z MEGA2560 jde na pin #12 posuvného registru Digitál 9 z MEGA2560 jde na pin #11 posuvného registru.

Všechny výstupy integrovaného obvodu kromě jednoho jsou na levé straně čipu. Proto jsou tam pro snadnější zapojení také LED diody.

Za čipem umístěte rezistory. Musíte dávat pozor, aby se žádné vodiče rezistorů vzájemně nedotýkaly. Před připojením napájení k MEGA2560 byste to měli ještě jednou zkontrolovat. Pokud je pro vás obtížné uspořádat rezistory tak, aby se jejich vývody nedotýkaly, pomůže zkrátit vývody tak, aby ležely blíže k povrchu desky.

Poté umístěte LED diody na destičku. Všechny delší kladné vodiče LED musí směřovat k čipu, ať už jsou na kterékoli straně desky.

Připojte propojovací vodiče, jak je znázorněno výše. Nezapomeňte na ten, který vede z pinu 8 integrovaného obvodu do sloupce GND na desce.

Nahrajte o něco později uvedený náčrt a vyzkoušejte jej. Každá LED dioda by měla postupně svítit, dokud se nerozsvítí všechny LED diody, a pak všechny zhasnou a cyklus se opakuje.

Kód

Po zapojení otevřete program ve složce s kódem - Lekce 11 Osm LED s

74HC595 a klikněte na tlačítko UPLOAD pro nahrání programu. Podrobnosti o nahrávání programu naleznete v lekci 2, pokud se vyskytnou chyby.

Nejprve definujeme tři piny, které budeme používat. Jedná se o digitální výstupy MEGA2560, které budou připojeny k latch, hodinovým a datovým pinům obvodu 74HC595.

```
int latchPin = 11;
```

```
int clockPin = 9;
```

```
int dataPin = 12;
```

Dále je definována proměnná 'leds'. V této proměnné se bude uchovávat vzor toho, které LED diody jsou právě zapnuté nebo vypnuté. Data typu "byte" reprezentují čísla pomocí osmi bitů. Každý bit může být buď zapnutý, nebo vypnutý, takže je to

ideální pro sledování počtu bitů.

www.elegoo.cz

které z našich osmi LED diod jsou

zapnuté nebo vypnuté. `byte leds = 0;`

Funkce 'setup' pouze nastaví tři piny, které používáme, jako digitální výstupy.

```
void setup()
```

```
{  
    pinMode(latchPin, OUTPUT);  
    pinMode(dataPin, OUTPUT);  
    pinMode(clockPin, OUTPUT);  
}
```

Funkce 'loop' nejprve vypne všechny LED diody tím, že proměnné 'leds' přiřadí hodnotu 0. Poté zavolá funkci "updateShiftRegister", která pošle vzor "leds" do posuvného registru, takže všechny LED zhasnou. Jak funguje funkce 'updateShiftRegister', se budeme zabývat později.

Funkce smyčky se na půl sekundy zastaví a poté začne počítat od 0 do 7 pomocí smyčky 'for' a proměnné 'i'. Pokaždé použije funkci Arduino 'bitSet' k nastavení bitu, který ovládá danou LED diodu v proměnné 'leds'. Pak také zavolá funkci 'updateShiftRegister', aby se ledky aktualizovaly podle toho, co je v proměnné 'leds'. Poté dojde k půlvteřinové prodlevě, než se inkrementuje 'i' a rozsvítí se další LED. `void loop()`

```
{  
    leds = 0;  
    updateShiftRegister();  
    delay(500);  
    for (int i = 0; i < 8; i++)  
    {  
        bitSet(leds, i);  
        updateShiftRegister();  
        delay(500);  
    }  
}
```

Funkce 'updateShiftRegister' nejprve nastaví latchPin na nízkou hodnotu, poté zavolá funkci MEGA2560 'shiftOut' a poté opět nastaví latchPin na vysokou hodnotu. Tato funkce přijímá čtyři parametry, přičemž první dva jsou piny, které se mají použít pro Data, respektive Clock. Třetí parametr určuje, od kterého konce dat se má začít. My začneme pravým bitem, který se označuje jako "

icant".

www.elegoo.cz

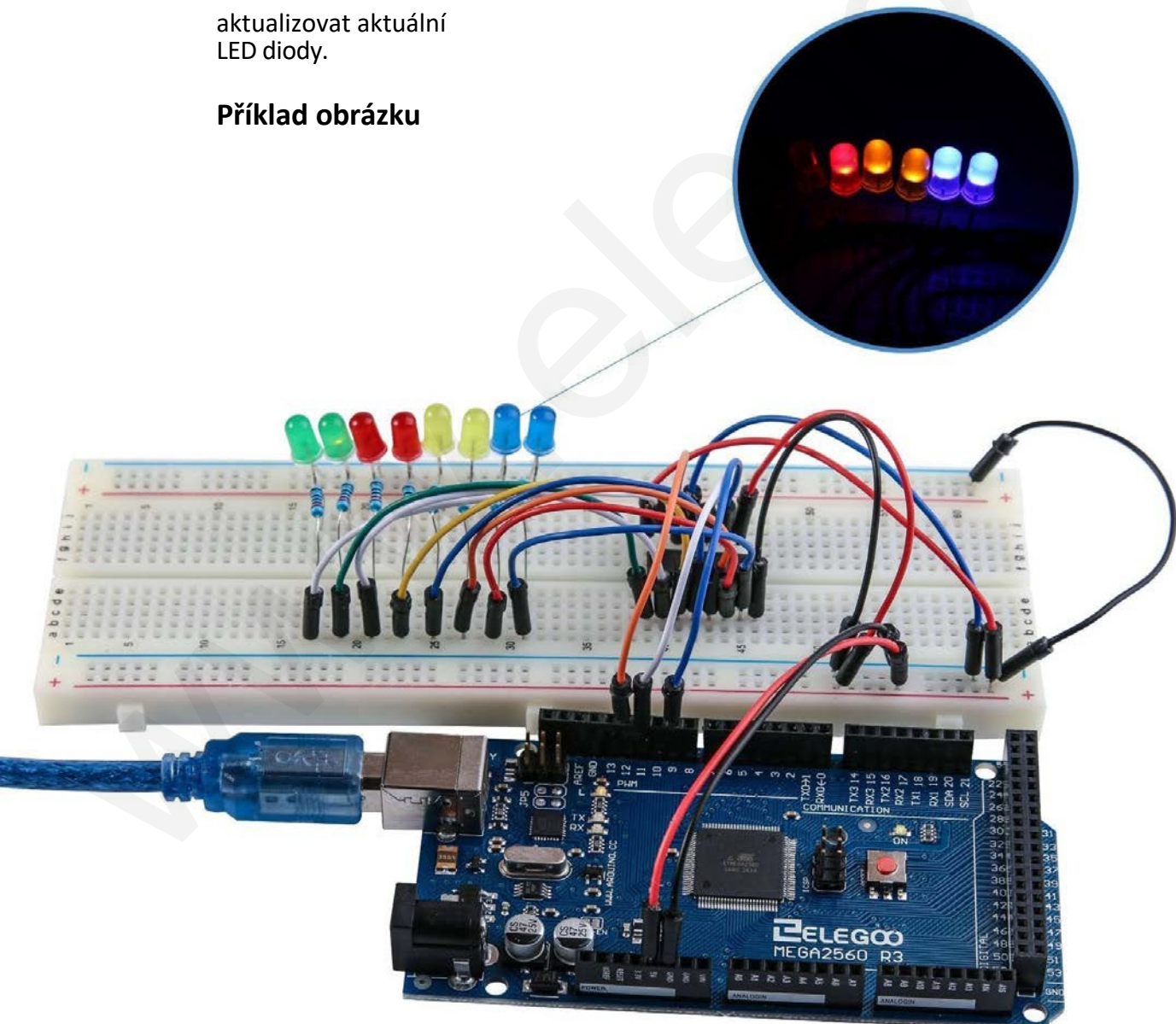
Bit' (LSB).

Posledním parametrem jsou skutečná data, která mají být posunuta do posuvného registru, což je v tomto případě 'leds'.

```
void updateShiftRegister()
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, leds);
    digitalWrite(latchPin, HIGH);
}
```

Pokud byste chtěli některou z LED diod vypnout a ne zapnout, zavolali byste podobnou funkci Arduino (bitClear) s proměnnou 'leds'. Tím se nastaví bit 'leds' na hodnotu 0 a pak by stačilo zavolat funkci 'updateShiftRegister', aby se aktualizoval aktuální LED diody.

Příklad obrázku



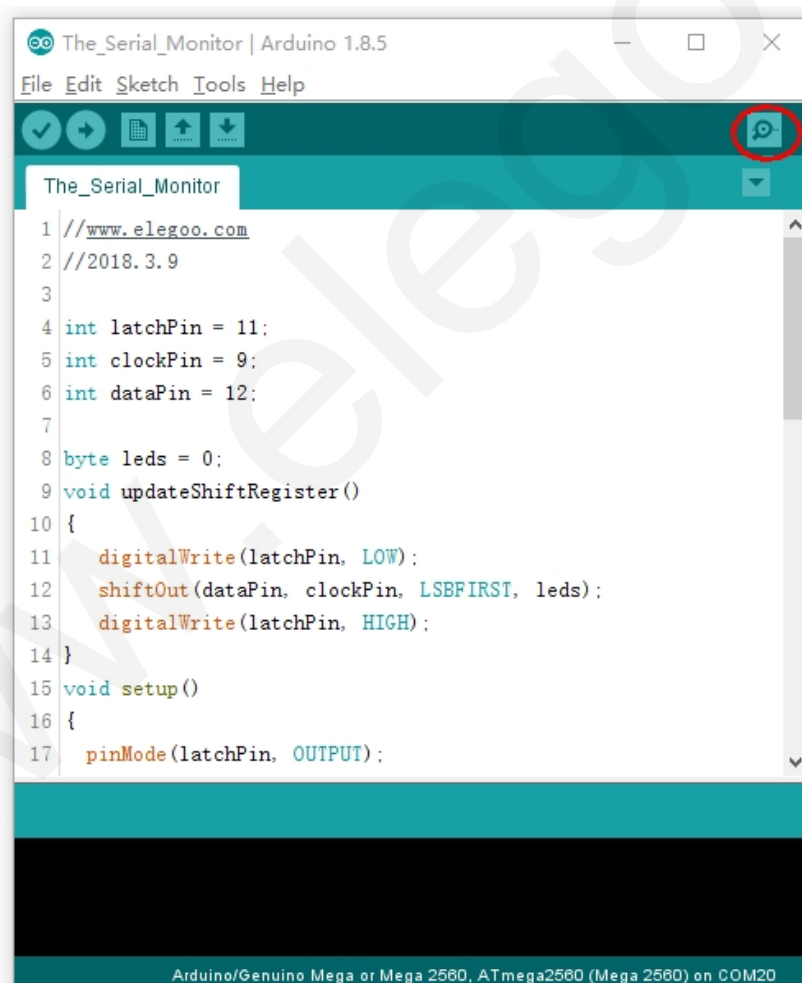
Lekce 12 Sériový monitor

Přehled

V této lekci navážete na lekci 11 a přidáte možnost ovládat LED diody z počítače pomocí sériového monitoru Arduino. Sériový monitor je "pojítkem" mezi počítačem a vaším MEGA2560. Umožňuje odesílat a přijímat textové zprávy, což se hodí při ladění a také při ovládání MEGA2560 z klávesnice! Například budete moci z počítače odesílat příkazy k zapnutí LED diod.

V této lekci použijete přesně stejné součástky a podobné uspořádání desky jako v lekci 11. Pokud jste tak ještě neučinili, postupujte nyní podle lekce 11.

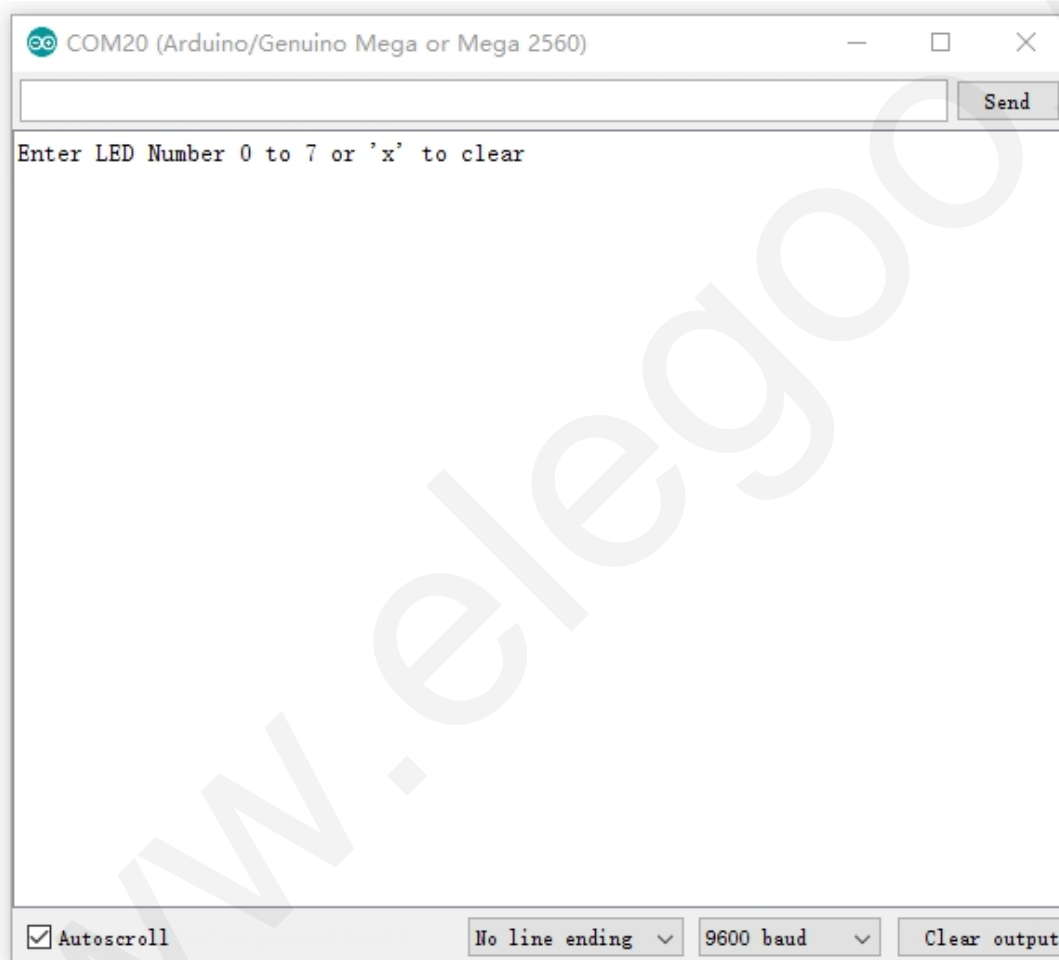
Přijaté kroky



Po nahrání tohoto náčrtu do MEGA2560 klikněte na pravé tlačítko na panelu nástrojů v Arduino IDE. Tlačítko je zakroužkováno níže.

Otevře se následující okno.

Kliknutím na tlačítko Serial Monitor zapněte sériový monitor. Základní informace o sériovém monitoru jsou podrobně představeny v lekcí 1.

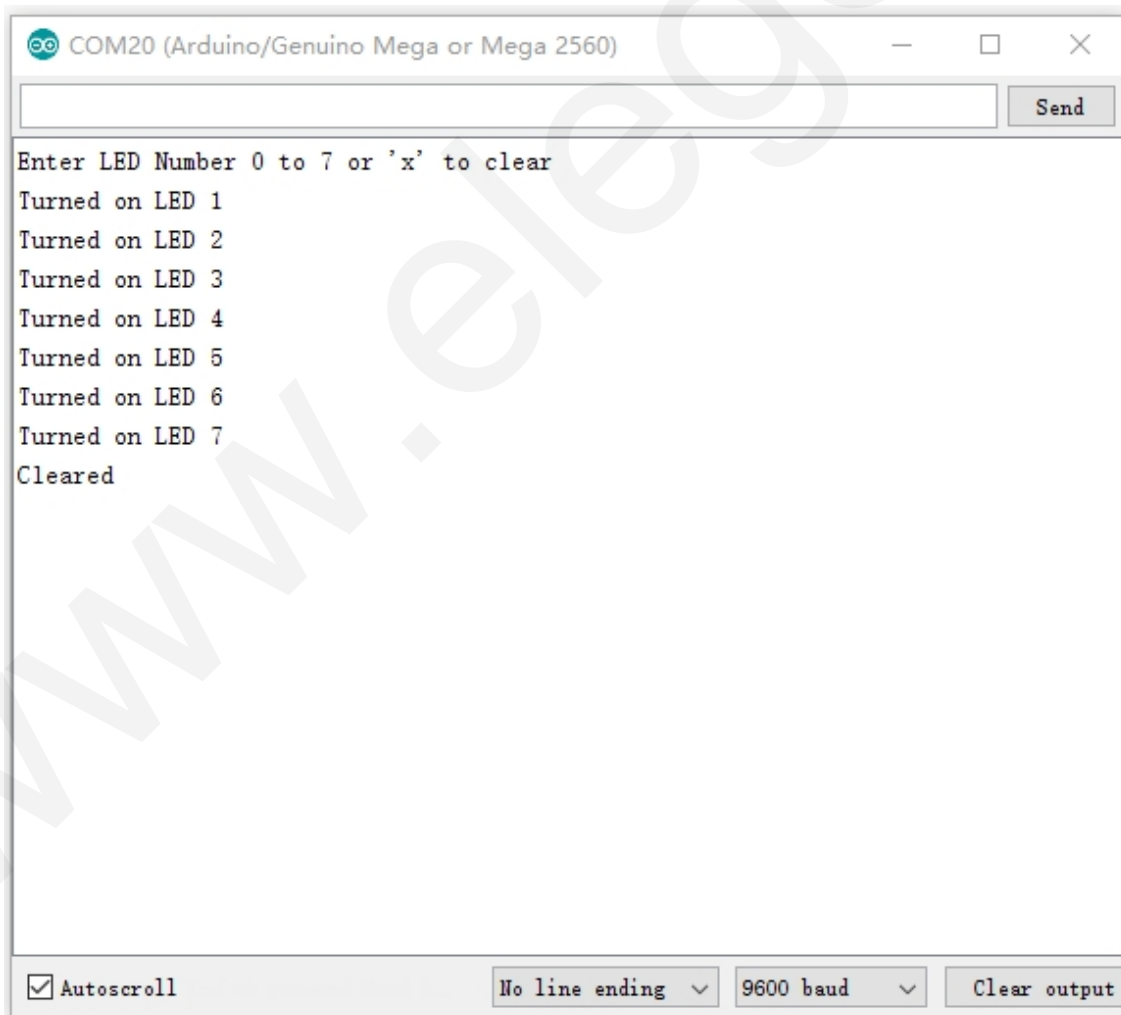


Toto okno se nazývá Serial Monitor a je součástí softwaru Arduino IDE. Jeho úkolem je umožnit odesílání zpráv z počítače na desku MEGA2560 (přes USB) a také přijímání zpráv z desky MEGA2560.

Arduino odeslalo zprávu "Zadejte číslo LED 0 až 7 nebo 'x' pro vymazání". Říká nám, jaké příkazy můžeme Arduino poslat: buď pošlete 'x' (pro vypnutí všech LED), nebo číslo LED, kterou chcete zapnout (kde 0 je spodní LED, 1 je další nahoře, až po 7 pro horní LED).

Zkuste zadat následující příkazy do horní oblasti sériového monitoru, která je na úrovni tlačítka "Odeslat". Po zadání každého z těchto znaků stiskněte tlačítko "Odeslat": x 0 3 5

Zadání x nebude mít žádný účinek, pokud jsou již všechny LED diody zhasnuté, ale po zadání každého čísla by se měla příslušná LED dioda rozsvítit a z desky MEGA2560 se zobrazí potvrzovací zpráva. Sériový monitor se zobrazí podle



následujícího obrázku.

www.elegoo.cz

Opět zadejte x a stiskněte tlačítko "Odeslat", čímž vypnete všechny kontrolky LED.

Kód

Po zapojení otevřete program ve složce s kódem - [Lekce 12 Sériový monitor a kliknutím na UPLOAD program nahrajte](#). Podrobnosti o nahrávání programu naleznete v [Lekci 2](#), pokud se vyskytnou chyby.

Jak se dalo očekávat, nákres vychází z nákresu použitého v lekci 24. Proto se zde budeme zabývat pouze novými částmi. Určitě se vám bude hodit odkaz na celý náčrtek v prostředí Arduino IDE.

Ve funkci 'setup' jsou na konci tři nové řádky: `void setup()`

```
{  
  pinMode(latchPin, OUTPUT);  
  pinMode(dataPin, OUTPUT);  
  pinMode(clockPin, OUTPUT);  
  updateShiftRegister();  
  Serial.begin(9600);  
  
  while (! Serial); // Počkejte, až bude Serial připraven -  
  Leonardo Serial.println("Zadejte číslo LED 0 až 7 nebo 'x' pro  
  vymazání");  
}
```

Nejprve máme příkaz 'Serial.begin(9600)'. Tím se spustí sériová komunikace, takže MEGA2560 může odesílat příkazy prostřednictvím připojení USB. Hodnota 9600 se nazývá "přenosová rychlost" připojení. Jedná se o rychlost, jakou se mají data odesílat. Tuto hodnotu můžete změnit na vyšší, ale budete muset také změnit monitor Arduino Serial na stejnou hodnotu. O tom si povíme později; prozatím ji ponechte na hodnotě 9600.

Řádek začínající 'while' zajišťuje, že na druhém konci připojení USB je něco, s čím může Arduino komunikovat, než začne odesílat zprávy. Jinak by se mohlo stát, že zpráva bude odeslána, ale nebude zobrazena. Tento řádek je ve skutečnosti nutný pouze v případě, že používáte Arduino Leonardo, protože Arduino MEGA2560 automaticky resetuje desku Arduino, když otevřete Serial Monitor, zatímco v případě Arduina Leonardo se tak neděje.

Poslední z nových řádků v 'setup' odešle zprávu, kterou vidíme v horní části Serial Monitoru.

Funkce 'loop' je místem, kde se odehrává veškerá akce: `void loop()`

```
{
  if (Serial.available())
  {
    char ch = Serial.read();
    if (ch >= '0' && ch <= '7')
    {
      int led = ch - '0';
      bitSet(leds, led);
      updateShiftRegister();
      Serial.print("Zapnuto LED ");
      Serial.println(led);
    }
    if (ch == 'x')
    {
      leds = 0; updateShiftRegister();

      Serial.println("Vymazáno");
    }
  }
}
```

Vše, co se děje uvnitř smyčky, je obsaženo v příkazu 'if'. Pokud tedy volání vestavěné funkce Arduina 'Serial.available()' není 'true', nic dalšího se nestane.

Serial.available() vrátí 'true', pokud byla data odeslána do MEGA2560 a jsou připravena ke zpracování. Příchozí zprávy jsou uchovávány v tzv. vyrovnávací paměti a Serial.available() vrací true, pokud je tato vyrovnávací paměť neprázdná.

Pokud byla zpráva přijata, přejde se na další řádek kódu: `char ch =`

`Serial.read();`

Tím se načte další znak z vyrovnávací paměti a odstraní se z vyrovnávací paměti.

Přiřadí jej také do proměnné 'ch'. Proměnná 'ch' je typu 'char', což znamená 'znak', a jak název napovídá, obsahuje jeden znak.

Pokud jste postupovali podle pokynů ve výzvě v horní části sériového monitoru, bude tento znak buď jednociferné číslo v rozmezí 0 až 7, nebo písmeno 'x'.

Příkaz 'if' na dalším řádku zkontroluje, zda se jedná o jednočíslici, a to tak, že zjistí, zda je 'ch' větší nebo rovno znaku '0' a menší nebo rovno znaku '7'. Takové porovnávání znaků vypadá trochu zvláštně, ale je naprosto přijatelné.

Každý znak je reprezentován jedinečným číslem, které se nazývá hodnota ASCII. To znamená, že když porovnáváme znaky pomocí `<=` a `>=`, porovnávají se ve skutečnosti hodnoty ASCII.

Pokud test proběhne úspěšně, přejdeme na další řádek:

`int led = ch - '0';`

Nyní provádíme aritmetiku na znacích! Odečítáme číslici '0' od zadané číslice.

Pokud jste tedy zadali '0', pak se '0' - '0' bude rovnat 0. Pokud jste zadali '7', pak se '7' - '0' bude rovnat číslu 7, protože ve skutečnosti jde o hodnoty ASCII, které se používají při odečítání.

Protože známe číslo LED diody, kterou chceme zapnout, stačí nastavit tento bit v proměnné 'leds' a aktualizovat posuvný registr.

`bitSet(leds, led);`

`updateShiftRegister();`

Následující dva řádky odešlou zpět potvrzovací zprávu do Serial Monitoru.

```
Serial.print("Zapnuto LED ");
```

```
Serial.println(led);
```

První řádek používá Serial.print místo Serial.println. Rozdíl mezi nimi je v tom, že Serial.print nezačíná nový řádek po vypsání toho, co je v jeho parametru. Toho využíváme v prvním řádku, protože tiskneme zprávu ve dvou částech. Nejprve obecnou část: Poté se zobrazí číslo LED diody.

Číslo LED diody je uloženo v proměnné 'int', nikoli v textovém řetězci. Soubor Serial.print může přijmout buď textový řetězec uzavřený v uvozovkách, nebo proměnnou 'int' nebo v podstatě jakýkoli typ proměnné.

Po příkazu 'if', který zpracovává případ, kdy byla zpracována jedna číslice, následuje druhý příkaz 'if', který zjišťuje, zda 'ch' je písmeno 'x'.

```
if (ch == 'x')
{
    leds = 0; updateShiftRegister();
    Serial.println("Vymazáno");
}
```

Pokud ano, vymaže všechny LED diody a odešle potvrzovací zprávu.

Lekce 13 Fotobuňka

Přehled

V této lekci se naučíte měřit intenzitu světla pomocí analogového vstupu. Navážete na lekci 12 a pomocí úrovně světla budete řídit počet rozsvícených LED diod.

Fotobuňka je na spodní straně desky, kde byl nahoře hrnec.

Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x 830 vázacích bodůdeska na chleba
- (8) x LED diody
- (8) x 220 ohmových rezistorů
- (1) x 1k ohmový odpor
- (1) x 74hc595 IC
- (1) x fotorezistor (fotobuňka)
- (16) x vodiče M-M (propojovací vodiče typu Male to Male)



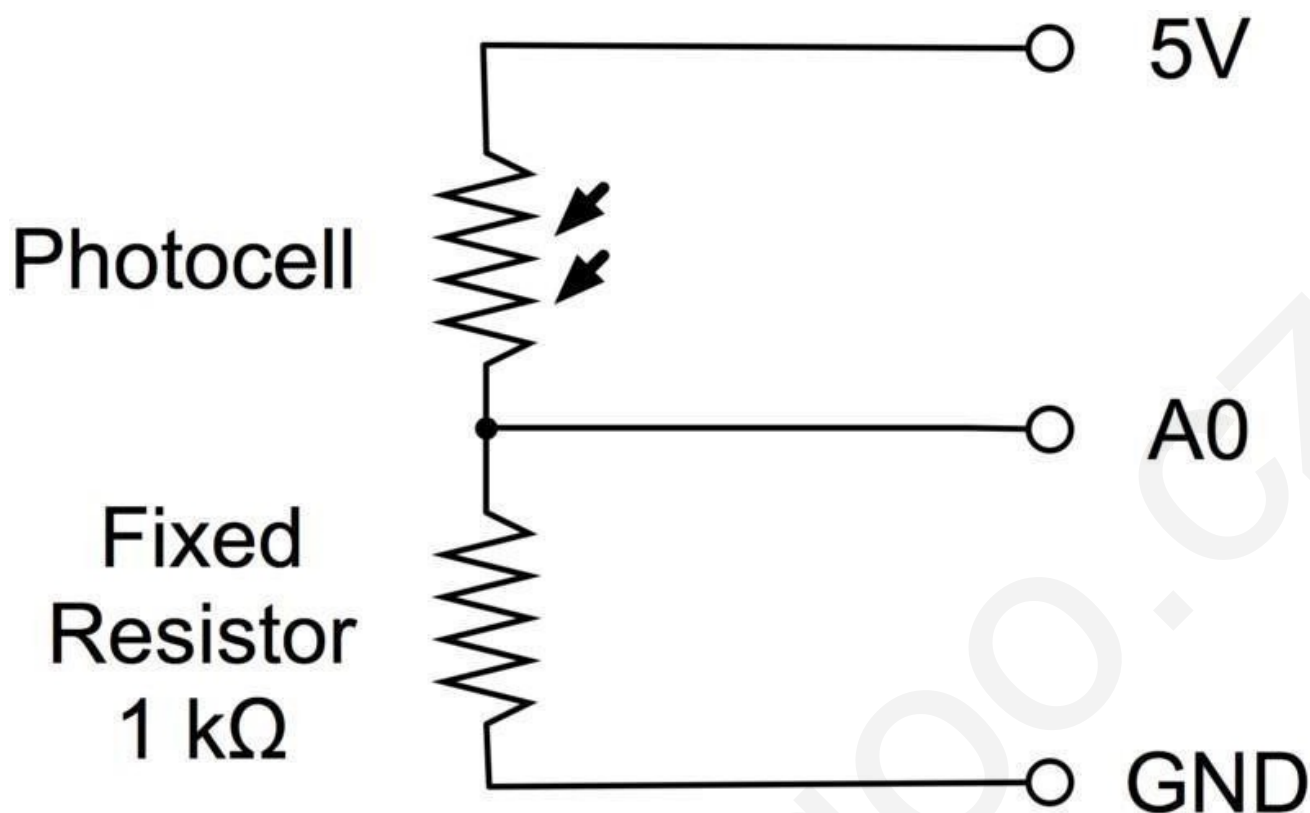
Úvod do složky

FOTOCCELL:

Použitý fotobuňka je typu zvaného světelně závislý rezistor, někdy nazývaný LDR. Jak název napovídá, tyto součástky se chovají stejně jako rezistory s tím rozdílem, že jejich odpor se mění v závislosti na množství světla, které na ně dopadá.

Ten má odpor asi 50 k Ω v téměř tmě a 500 Ω na jasném světle. Abychom mohli tuto proměnlivou hodnotu odporu převést na něco, co můžeme změřit na analogovém vstupu desky MEGA 2560 R3, je třeba ji převést na napětí.

Nejjednodušší způsob, jak toho dosáhnout, je kombinovat jej s pevným odporem.



Rezistor a fotočlánek se společně chovají jako potenciometr. Když je světlo velmi jasné, je odpor fotobuňky ve srovnání s rezistorem s pevnou hodnotou velmi malý, a tak je to, jako by byl potenciometr otočen na maximum.

Když je fotobuňka v matném světle, odpor je větší než pevný odpor 1 kΩ a je to, jako by byl potenciometr otočen směrem k GND.

Nahrajte si náčrtek uvedený v následující části a zkuste fotobuňku zakrýt prstem a poté ji podržet v blízkosti zdroje světla.

Připojení Schéma

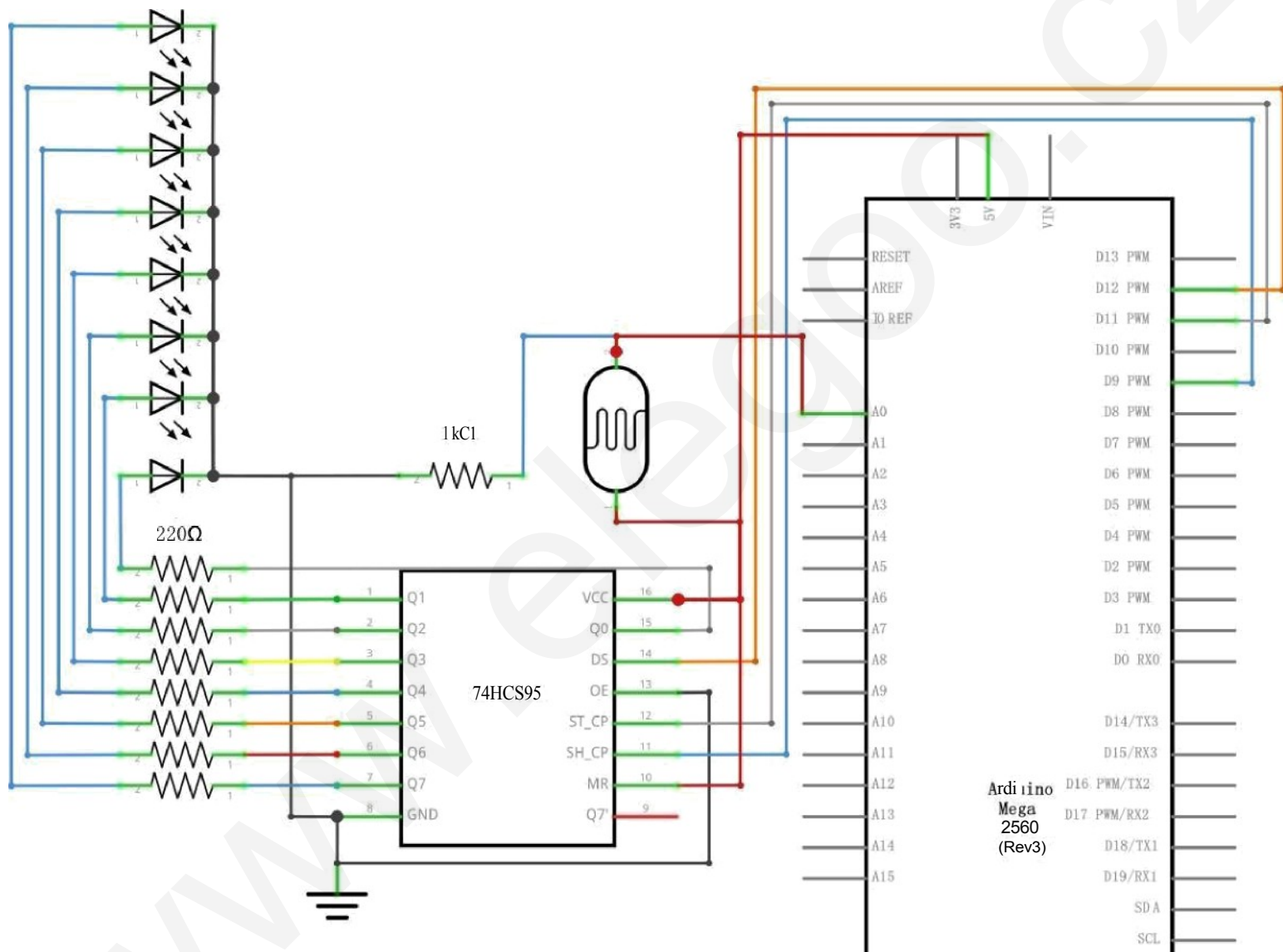
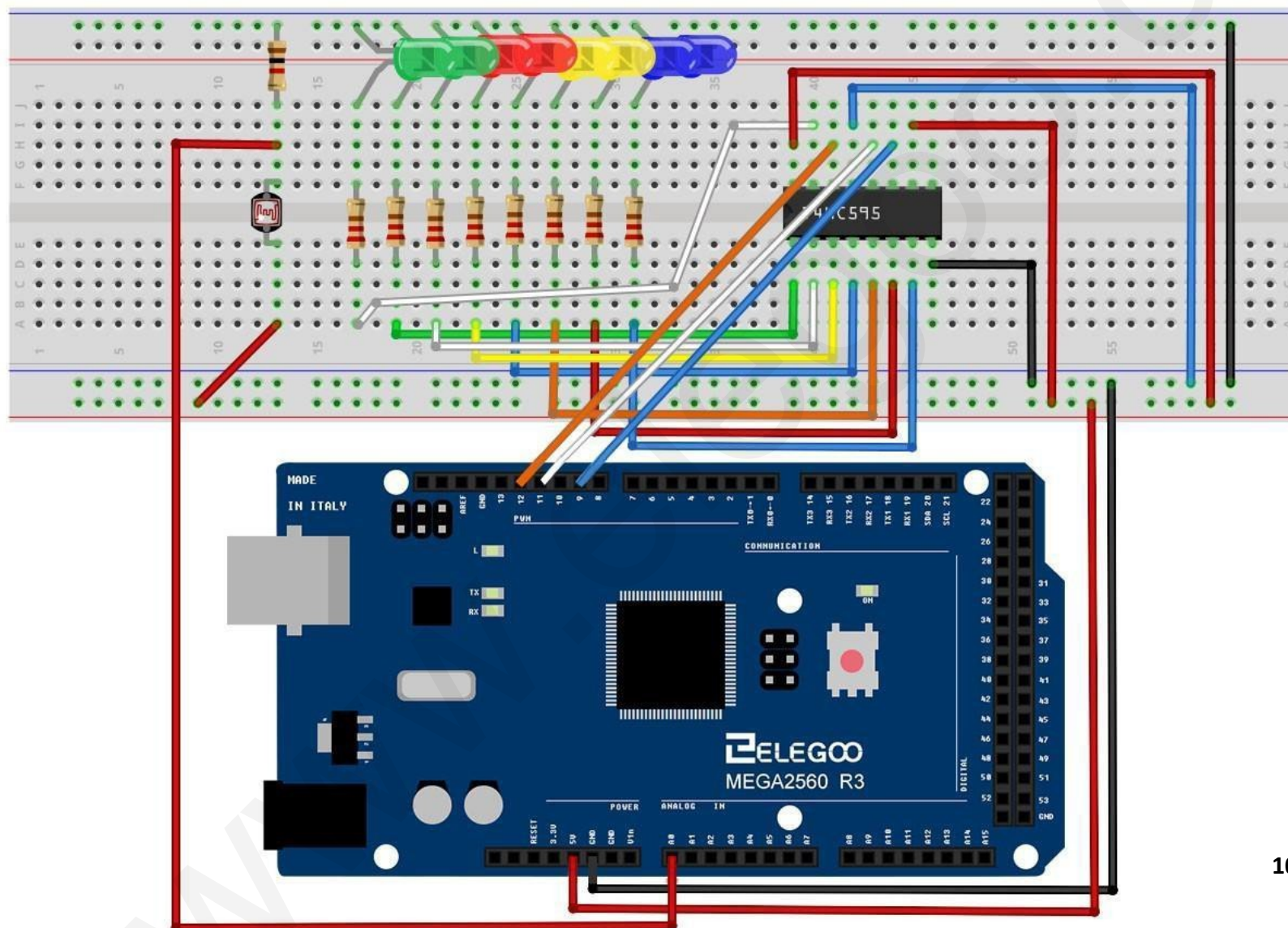


Schéma zapojení



Kód

Po zapojení otevřete program ve složce s kódem - Lekce 13 Fotobuňka a kliknutím na tlačítko **UPLOAD** nahrajte program. Podrobnosti o nahrávání programu naleznete v Lekci 2, pokud se vyskytnou chyby.

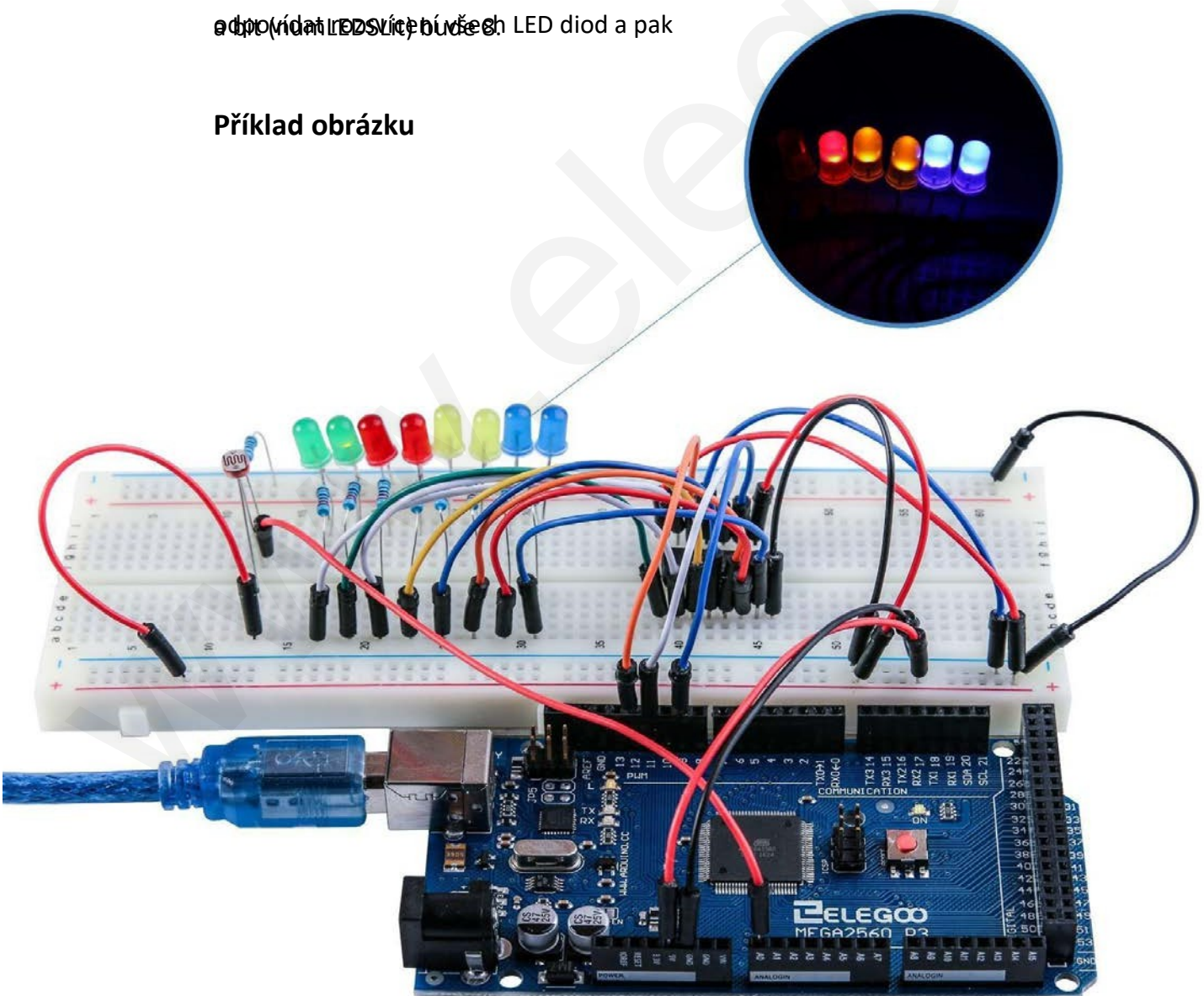
Nejdříve je třeba si všimnout, že jsme změnili název analogového pinu na "lightPin", nikoliv "potPin", protože již nemáme připojený potenciometr.

Jedinou další podstatnou změnou v náčrtu je řádek, který vypočítává, kolik LED diod se má rozsvítit:

```
int numLEDsLit = čtení / 57; // všechny LED svítí při 1k
```

Tentokrát vydělíme nezpracovanou hodnotu číslem 57 místo číslem 114. Jinými slovy, vydělíme jej o polovinu, jako jsme to udělali u hrnce, abychom jej rozdělili do devíti zón, od žádné rozsvícené LED diody po všech osm rozsvícených. Tento dodatečný faktor zohledňuje pevný odpor 1 kΩ. To znamená, že když má fotobuňka odpor 1 kΩ (stejný jako pevný odpor), hrubý údaj bude $1023 / 2 = 511$. To bude odpovídat rozsvícení všech LED diod a pak

Příklad obrázku



www.elegoo.cz

Lekce 14 Čtyři digitální sedmsegmentové displeje

Přehled

V této lekci se naučíte používat čtyřmístný sedmsegmentový displej.

Při použití 1místného 7segmentového displeje si prosím všimněte, že pokud se jedná o společnou anodu, připojuje se společný anodový kolík ke zdroji napájení; pokud se jedná o společnou katodu, připojuje se společný katodový kolík ke GND.

Při použití 4místného 7segmentového displeje se pro ovládání zobrazované číslice používá společný anodový nebo katodový kolík. Přestože pracuje pouze jedna číslice, princip perzistence vidění umožňuje vidět všechna zobrazená čísla, protože rychlost snímání každého z nich je tak vysoká, že si intervalů téměř nevšimnete.

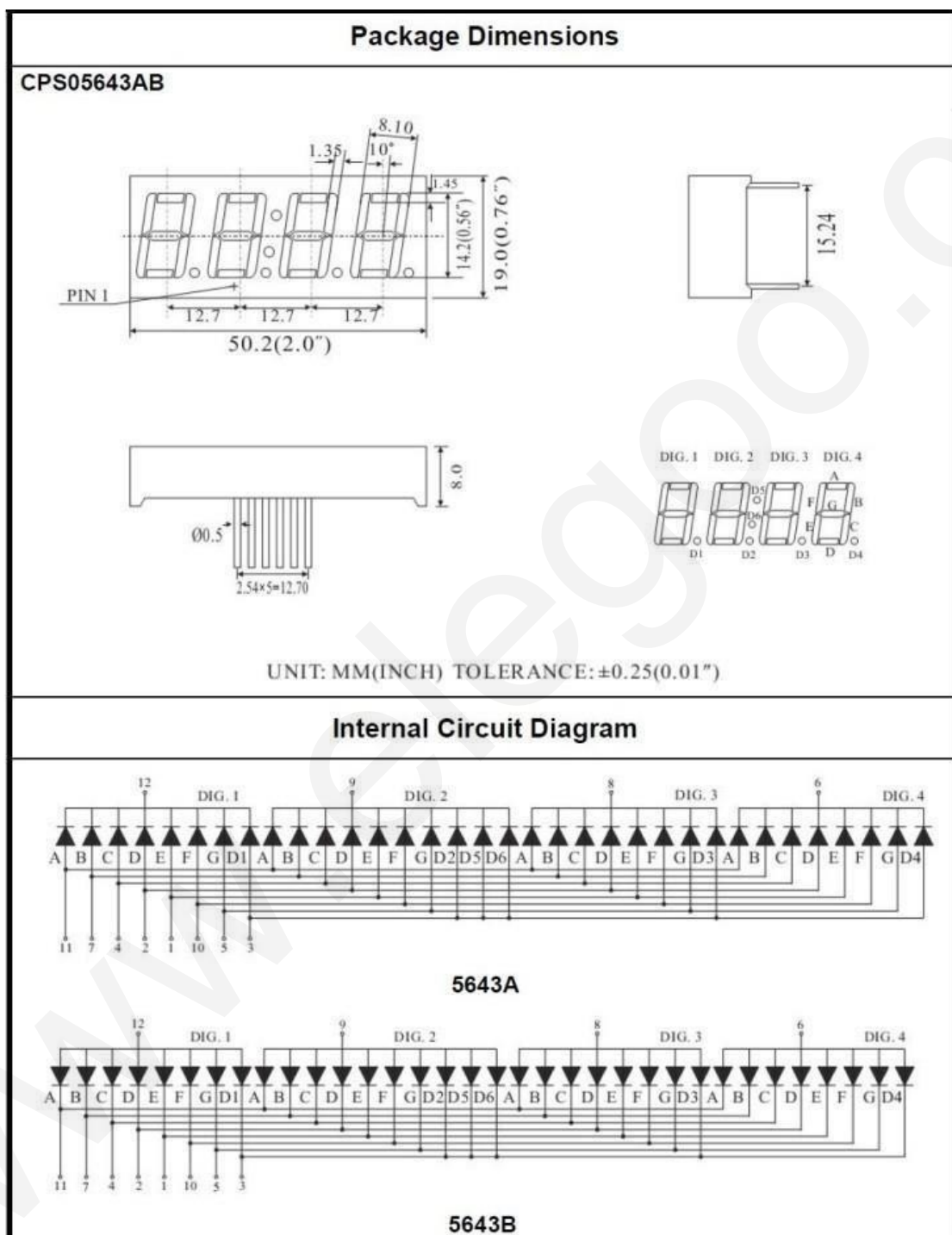
Požadovaná součást:

- (1) x Elegoo Mega 2560 R3
- (1) x 830 vázacích bodůdeska na chléb
- (1) x 74HC595 IC
- (1) x 4místný 7segmentový displej
- (4) x 220 ohmů rezistory
- (23) x vodiče M-M (propojovací vodiče typu Male to Male)



Úvod do složky

Čtyři digitální sedmisegmentové displeje



Four Digits Displays Series

Připojení Schéma

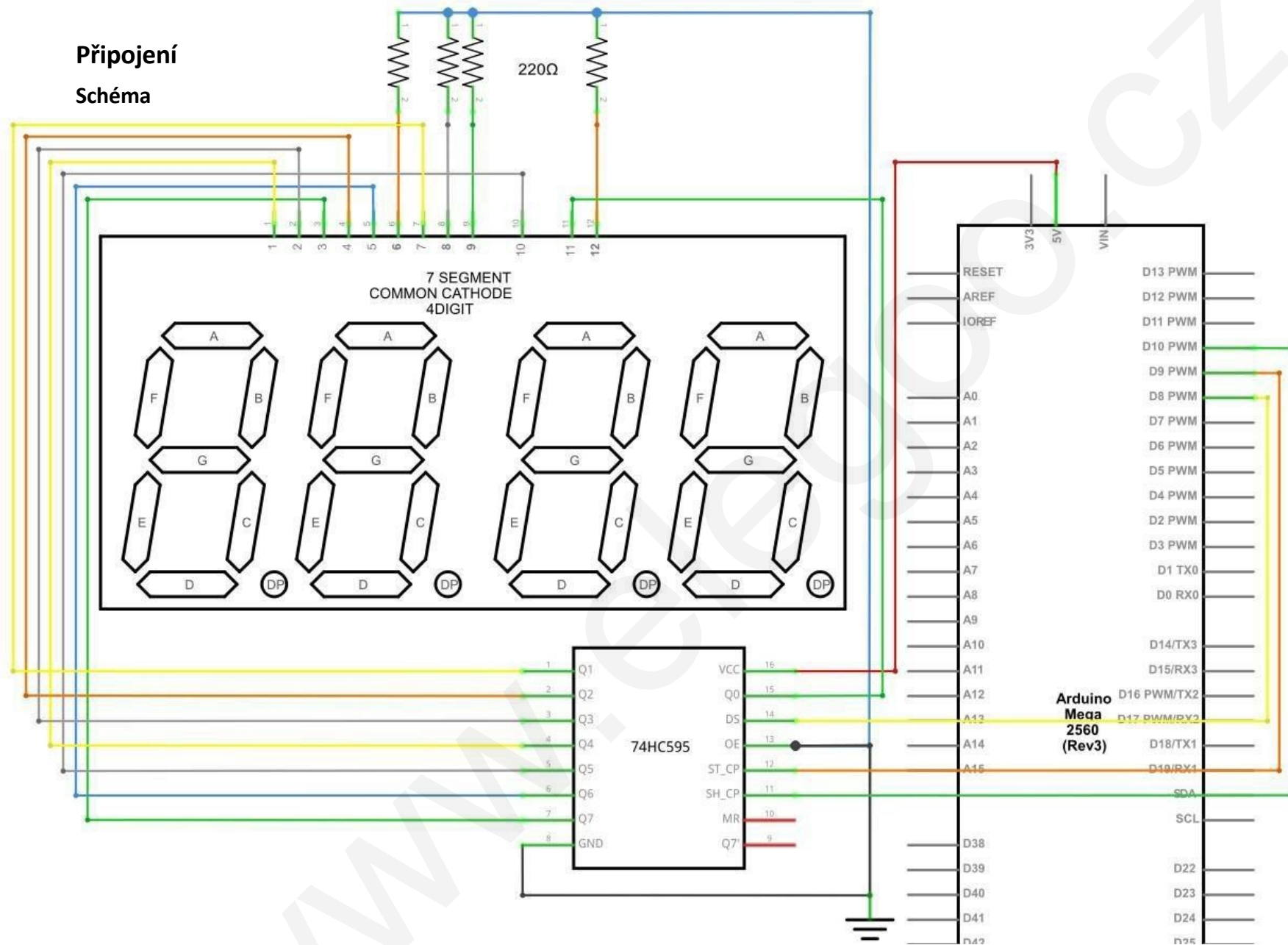
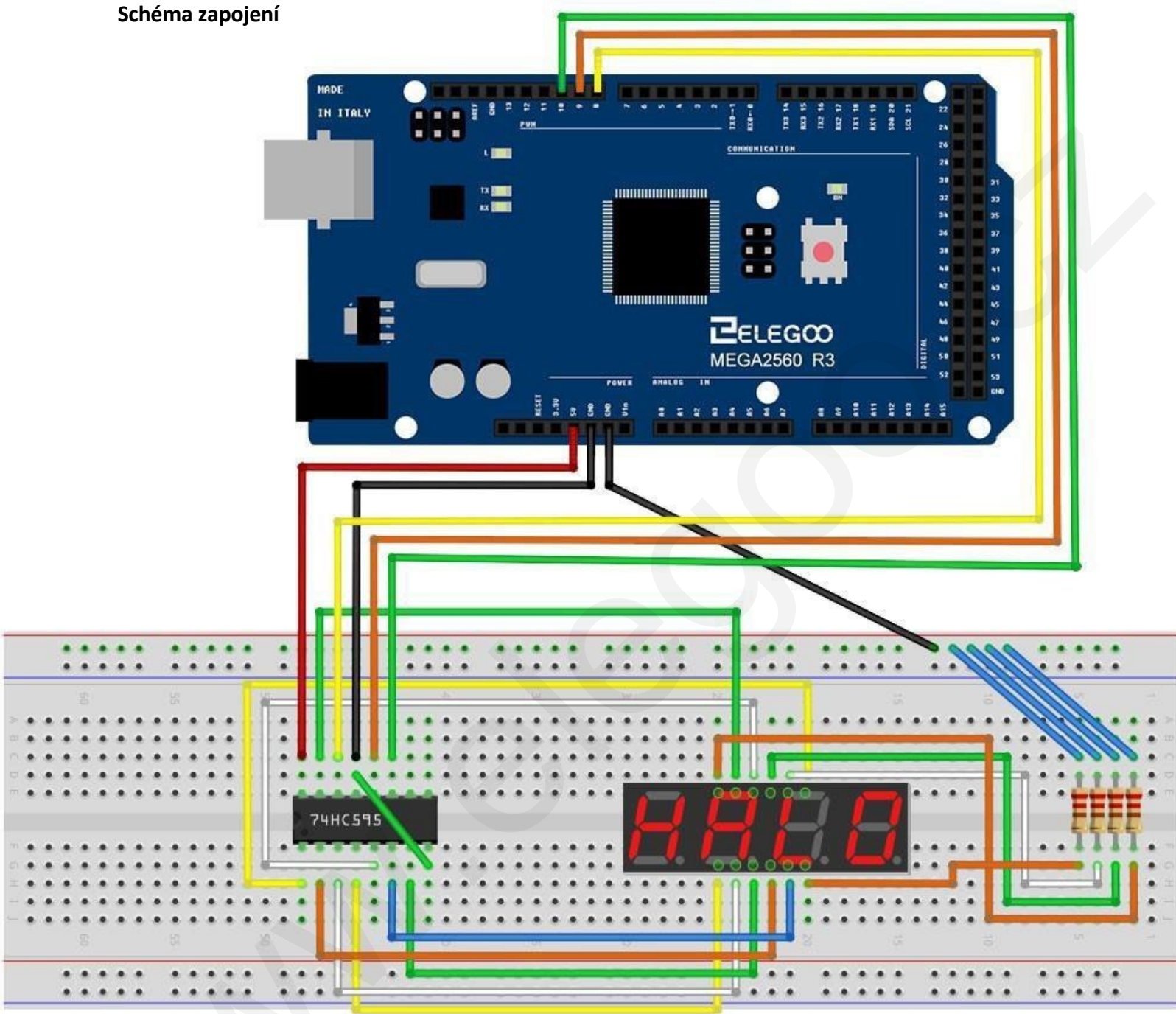


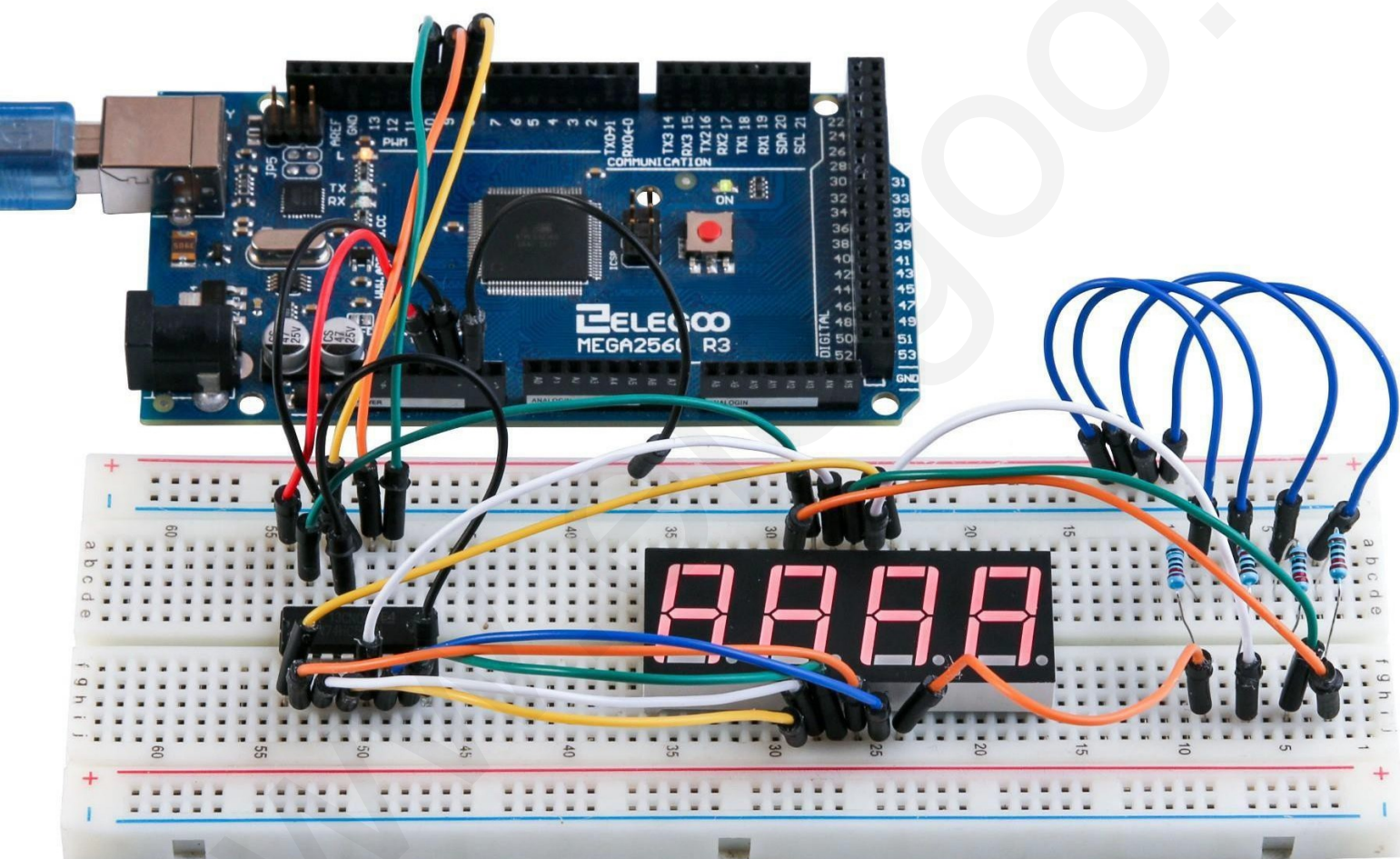
Schéma zapojení



Kód

Po zapojení otevřete program ve složce s kódem - Lekce 14 Čtyři digitální sedmisegmentové displeje a kliknutím na tlačítko UPLOAD nahrajte program. Podrobnosti o nahrávání programu naleznete v Lekci 2, pokud se vyskytnou nějaké chyby.

Příklad obrázku



Lekce 15 Stejnosměrné motory

Přehled

V této lekci se naučíte řídit malý stejnosměrný motor pomocí MEGA 2560 a tranzistoru.

Požadovaná součást:

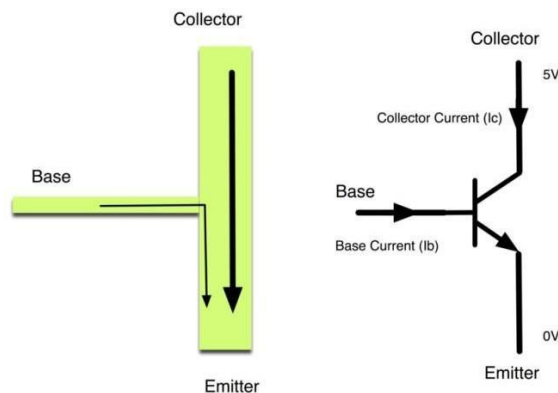
- (1) x Elegoo MEGA 2560
- (1) x Breadboard
- (1) x stejnosměrný motor 6V
- (1) x PN2222
- (1) x 1N4007
- (1) x 220 ohmů odpor
- (3) x vodiče M-M

Úvod do složky

TRANSISTORY

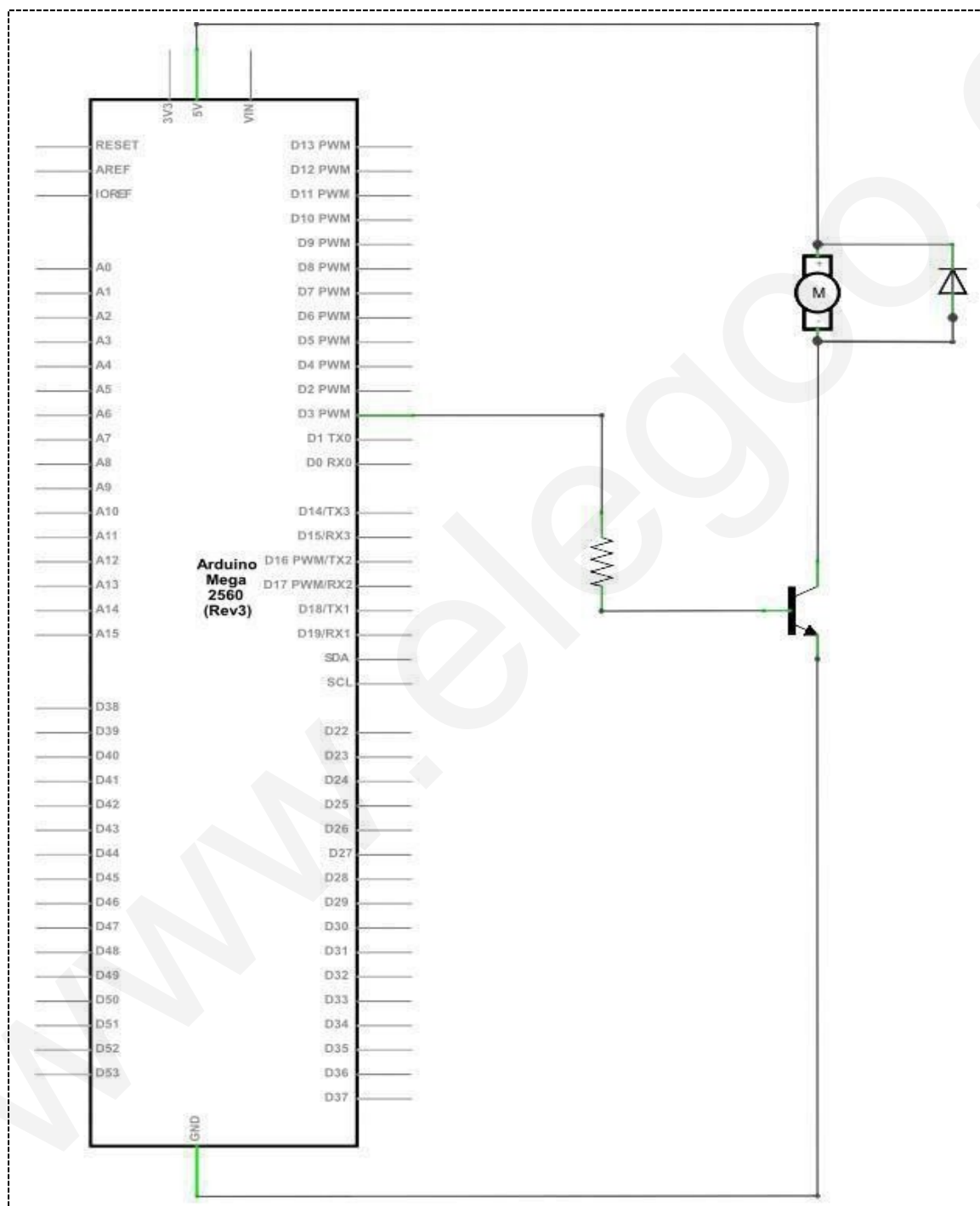
Malý stejnosměrný motor bude pravděpodobně spotřebovávat více energie, než kolik dokáže přímo zpracovat digitální výstup desky MEGA 2560. Pokud bychom se pokusili připojit motor přímo k vývodu desky MEGA 2560, existuje velká pravděpodobnost, že by mohlo dojít k poškození desky MEGA 2560.

Malý tranzistor, jako je PN2222, lze použít jako spínač, který využívá jen malý proud z digitálního výstupu desky MEGA 2560 k řízení mnohem většího proudu motoru.



Tranzistor má tři vývody. Většina elektřiny proudí z kolektoru do emitoru, ale to se stane pouze v případě, že malé množství proudu proudí do spoje báze. Tento malý proud je dodáván digitálním výstupem desky MEGA 2560.

Níže je uvedeno schéma tzv. Stejně jako schéma na desce je to způsob, jak znázornit, jak jsou součásti elektronického projektu vzájemně propojeny.



Vývod D3 desky MEGA2560 je připojen k rezistoru. Stejně jako při použití LED diody se tím omezí proud tekoucí do tranzistoru přes základnu.

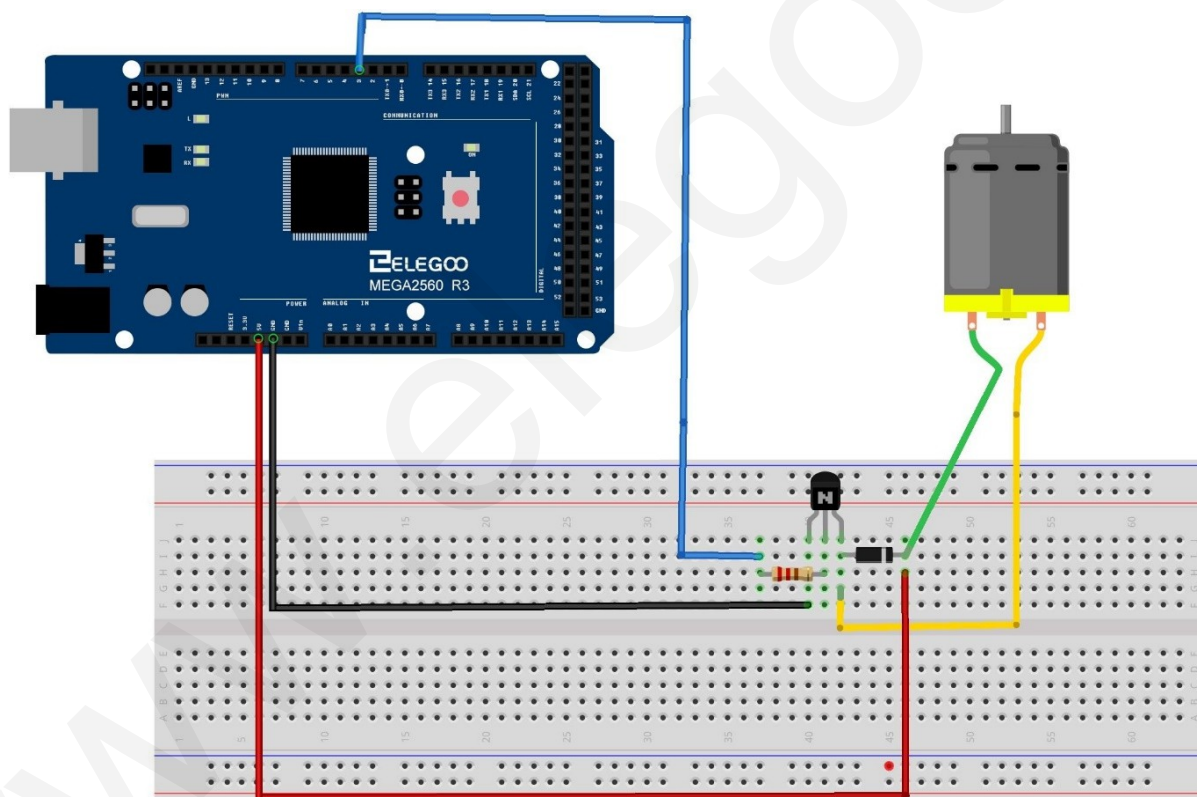
Přes spoje motoru je připojena dioda. Diody umožňují tok elektřiny pouze jedním směrem (ve směru jejich šíření).

Když vypnete napájení motoru, vznikne záporné napětí, které může poškodit desku MEGA2560 nebo tranzistor. Dioda proti tomu chrání tím, že zkratuje jakýkoli takový zpětný proud z motoru.

Tímto experimentem jste se naučili řídit LCD1602. Nyní můžete vytvářet vlastní zprávy, které se budou zobrazovat! Můžete také zkusit nechat LCD1602 zobrazovat čísla.

Připojení

schéma zapojení



K řízení otáček motoru budete používat analogový výstup desky MEGA2560 (PWM), a to odesláním čísla v rozsahu 0 až 255 ze sériového monitoru.

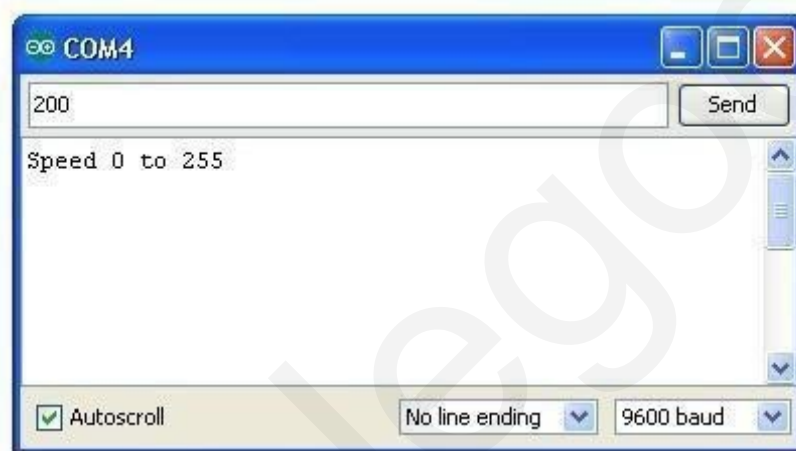
Při sestavování desky pro zapojení je třeba dbát na dvě věci.

Nejprve se ujistěte, že plochá strana tranzistoru je na pravé straně desky.

Za druhé, pruhovaný konec diody by měl směřovat k napájecímu vedení +5V - viz obrázek níže!

Motor dodávaný se stavebnicemi neodebírá více než 250 mA, ale pokud máte jiný motor, může snadno odebírat 1000 mA, což je více, než port USB zvládne! Pokud si nejste jisti proudovým odběrem motoru, napájejte desku MEGA 2560 ze síťového adaptéru, nikoli pouze z USB. Tranzistor funguje jako spínač a řídí napájení motoru, pin 3 desky MEGA 2560 slouží k zapínání a vypínání tranzistoru a ve skice je pojmenován "pin motoru".

Po spuštění skicu se zobrazí výzva, která vás upozorní, že pro ovládání rychlosti motoru je třeba zadat do SerialMonitoru hodnotu v rozmezí 0 až 255.

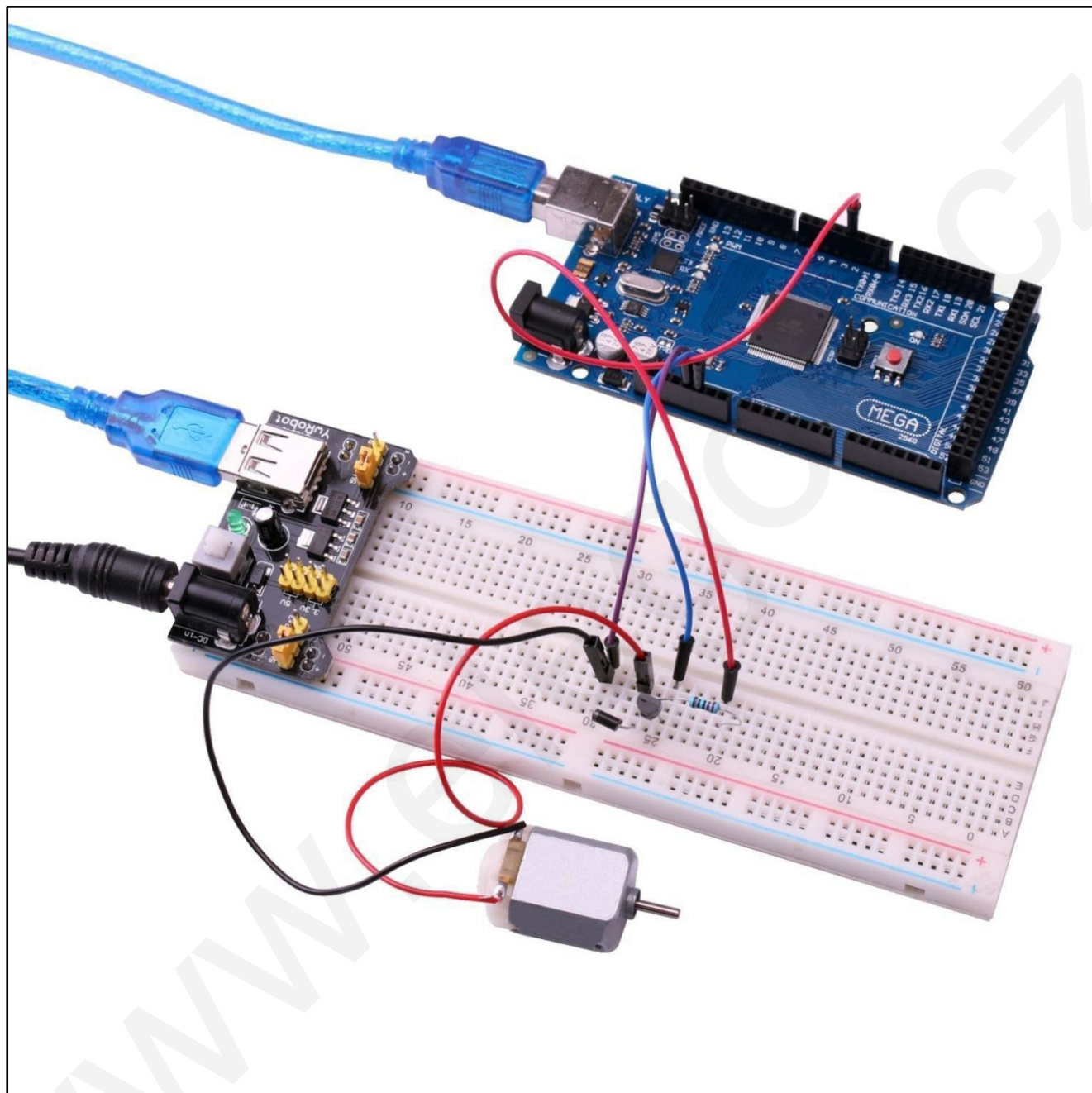


Ve funkci 'loop' se použije příkaz 'Serial.parseInt', který přečte číslo zadané jako text v Serial Monitoru a převede ho na 'int'.

Zde můžete zadat libovolné číslo, takže příkaz 'if' na dalším řádku provede analogový zápis s tímto číslem pouze tehdy, pokud je číslo v rozmezí 0 až 255.

www.elegoo.cz

Příklad obrázku



Lekce 16 Štafeta

Přehled

V této lekci se naučíte používat arelay.

Požadovaná součást:

- (1) x Elegoo MEGA 2560
- (1) x Breadboard
- (1) x PN2222
- (1) x 1N4007
- (1) x 220 ohmů
- (1) x stejnosměrný motor 6V
- (1) x relé
- (1) x napájecí modul na desce plošných spojů
- (1) x 9V napájecí adaptér
- (8) x vodiče M-M

Úvod do složky

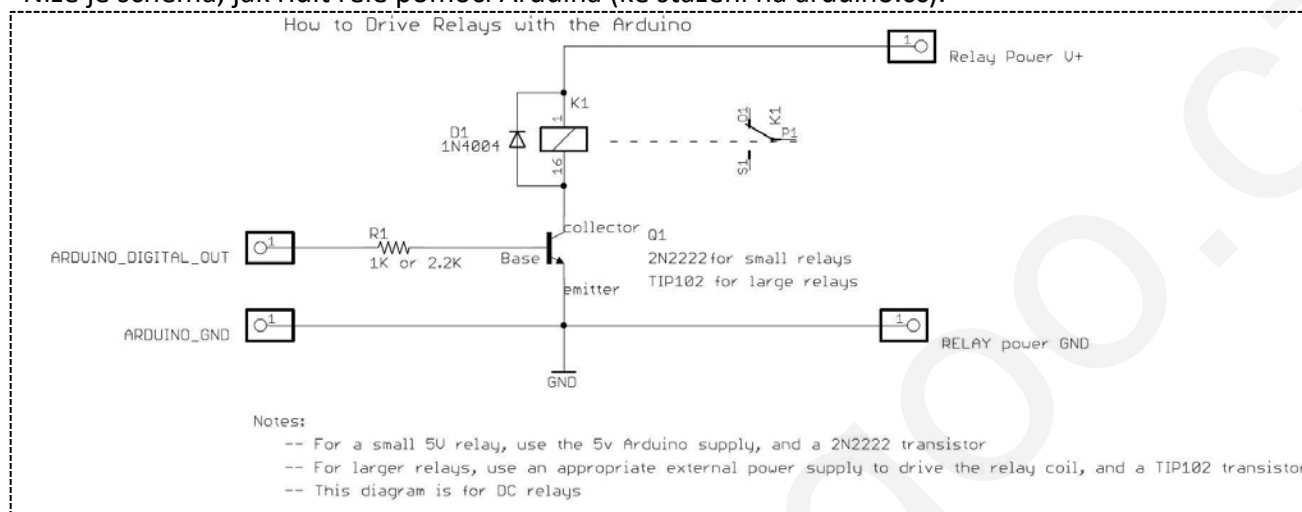
Štafeta:

Relé je elektricky ovládaný spínač. Mnoho relé používá elektromagnet k m e c h a n i c k é m u ovládní spínače, ale používají se i jiné principy ovládní jako u polovodičových relé. Relé se používají tam, kde je třeba ovládat obvod pomocí signálu s malým výkonem (s úplnou elektrickou izolací mezi řídicími a ovládanými obvody), nebo tam, kde je třeba ovládat několik obvodů jedním signálem. První relé se používala v dálkových telegrafních obvodech jako zesilovače. Opakovala signál přicházející z jednoho obvodu a znovu jej přenášela na jiný obvod. Relé se hojně používala v telefonních ústřednách a prvních počítačích k provádění logických operací.

Typ relé, který zvládne vysoký výkon potřebný k přímému ovládní elektromotoru nebo jiných zátěží, se nazývá stykač. Polovodičová relé ovládají silové obvody bez pohyblivých prvků.

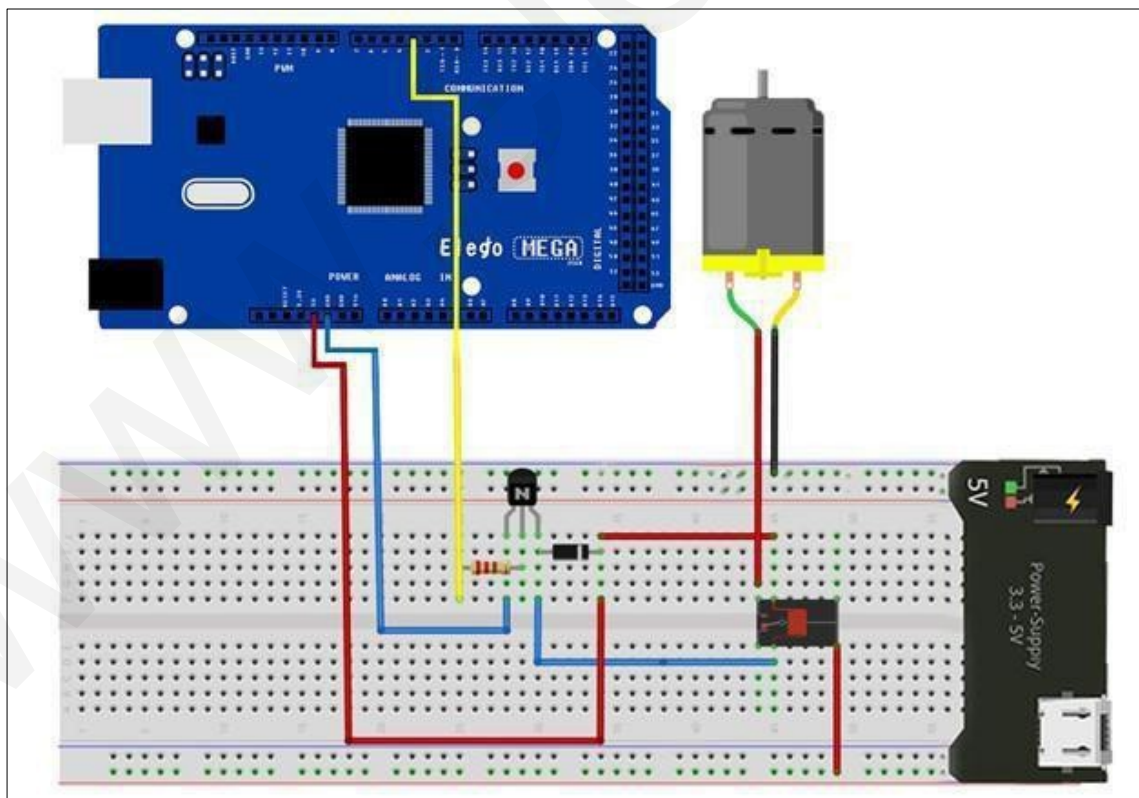
místo toho se k přepínání používá polovodičové zařízení. K ochraně elektrických obvodů před přetížením nebo poruchami se používají relé s kalibrovanými provozními charakteristikami a někdy s více provozními cívkami. V moderních elektroenergetických systémech tyto funkce vykonávají digitální přístroje nazývané "ochranná relé".

Níže je schéma, jak řídit relé pomocí Arduino (ke stažení na arduino.cc).



Připojení

schéma zapojení



Kód

Viz soubor s kódem.

