



**ZÁKLADNÍ VÝUKOVÝ PROGRAM
PRO STARTOVACÍ
SADU UNO**

Předmluva

Naše společnost

Společnost Elegoo Inc. byla založena v roce 2011 a je prosperující technologickou společností, která se zabývá výzkumem a vývojem, výrobou a marketingem open source hardwaru. Sídlíme v čínském Silicon Valley v Shenzhenu a rozrostli jsme se na více než 150 zaměstnanců s továrnou o rozloze více než 10 763 metrů čtverečních.

Naše produktové řady sahají od drátů DuPont, desek UNO R3 až po kompletní startovací sady určené pro zákazníky jakékoli úrovně, kteří se chtějí naučit ovládat Arduino. Kromě toho prodáváme také produkty příslušenství pro Raspberry Pi, jako jsou 2,8" dotykové TFT a STM32. V budoucnu bychom chtěli věnovat více energie a investic produktům pro 3D tiskárny apod. Všechny naše výrobky splňují mezinárodní standardy kvality a jsou velmi ceněny na různých trzích po celém světě.

Oficiální webové stránky: <http://www.elegoo.com>

US Amazon storefront: <http://www.amazon.com/shops/A2WWHQ25ENKVJ1>

CA Amazon storefront: <http://www.amazon.ca/shops/A2WWHQ25ENKVJ1>

UK Amazon storefront: <http://www.amazon.co.uk/shops/AZF7WYXU5ZANW>

DE Amazon storefront: <http://www.amazon.de/shops/AZF7WYXU5ZANW>

FR Amazon storefront: <http://www.amazon.fr/shops/AZF7WYXU5ZANW>

ES Amazon storefront: <http://www.amazon.es/shops/AZF7WYXU5ZANW>

IT Amazon storefront: <http://www.amazon.it/shops/AZF7WYXU5ZANW>

Náš výukový program

Tento návod je určen pro začátečníky. Dozvíte se všechny základní informace o tom, jak používat řídicí desku Arduino, senzory a komponenty. Pokud chcete studovat Arduino do větší hloubky, doporučujeme vám přečíst si knihu Arduino Cookbook, kterou napsal Michael Margolis.

Některé kódy v tomto návodu upravil Simon Monk. Simon Monk je autorem řady knih týkajících se open source hardwaru. Jsou k dispozici na Amazonu: Monk nabízí knihy: Programování Arduina, 30 projektů Arduina pro zlé génie a Programování Raspberry Pi.

Zákaznický servis

Jako neustále a rychle rostoucí technologická společnost se snažíme co nejlépe.

vám nabízíme vynikající produkty a kvalitní služby, které splní vaše očekávání, a můžete se na nás obrátit jednoduše na service@elegoo.com nebo EUservice@elegoo.com. Těšíme se na vaše odpovědi a jakákoli vaše kritická připomínka nebo návrh by pro nás byly velmi cenné.

Na všechny problémy a dotazy týkající se našich produktů vám naši zkušení technici okamžitě odpoví do 12 hodin (24 hodin během dovolené).

ack i n je



RGB LED
2 KS



Fotorezistor
(fotočlánek)
1KS



Rezistor
120ks



UNO R3
s USB

1
POČÍ
TAČ



Spínač
naklápěcí
koule

IPC



LED
30
KUSŮ



Knoflík(malý)
SPCS



Aktivní bzučák
1KS



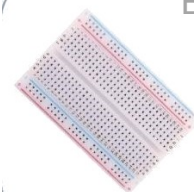
74HC595 IC
1KS



F- M Dupont Wire
SPCS



Propojovací
drát na desce
plošných spojů
65 KUSŮ



Breadboard
1KS

Obsah

Lekce 0 Instalace IDE	6
Lekce 1 Přidání knihoven a otevření sériového monitoru	17
Lekce 2 Mrknutí.....	26
Lekce 3 LED.....	37
Lekce 4 RGB LED	44
Lekce 5 Digitální vstupy	53
Lekce 6 Aktivní bzučák.....	58
Lekce 7 Spínač naklápěcí koule	62
Lekce 8 Osm LED s 74HC595.....	66
Lekce 9 Sériový monitor	73
Lekce 10 Fotobuňka.....	79

Lekce 0 Instalace prostředí IDE

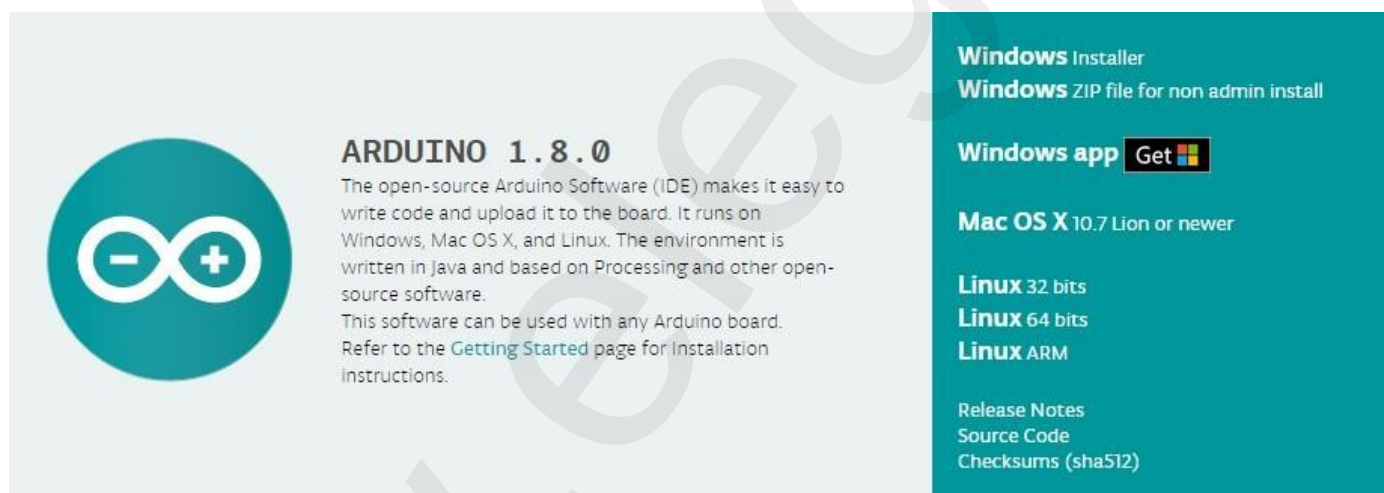
Úvod

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

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

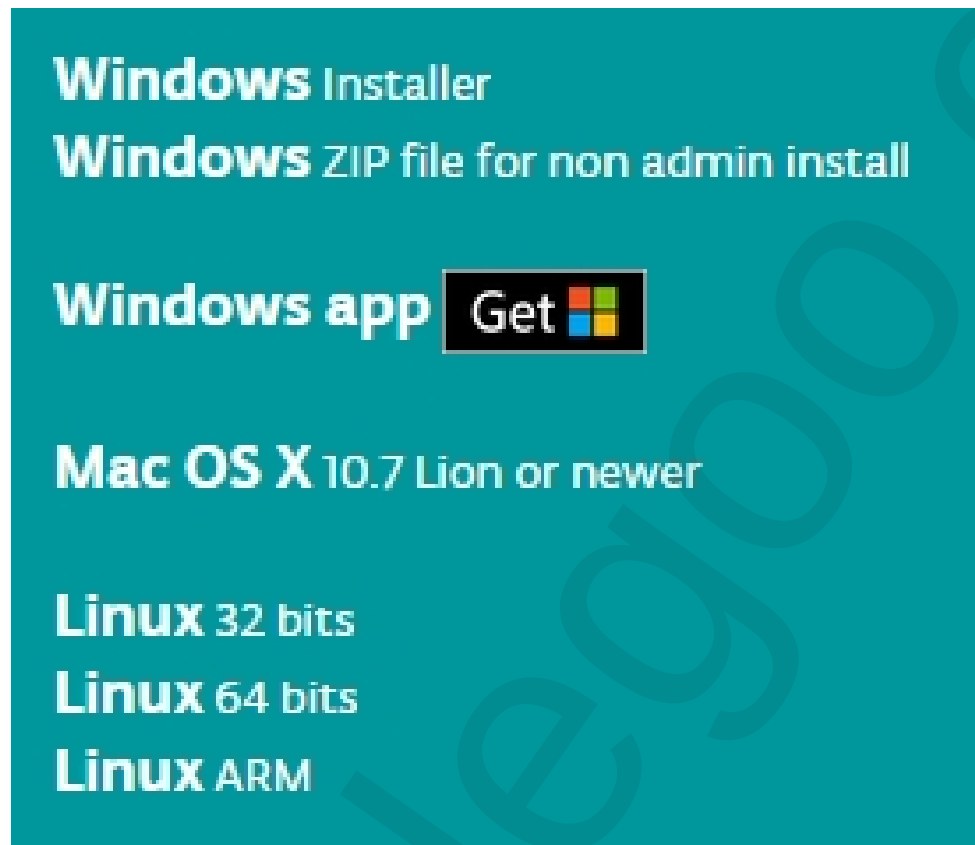
Software Arduino, který budete používat k programování Arduino, je k dispozici pro Windows, Mac a Linux. Instalační proces je pro všechny tři platformy odlišný a bohužel je nutné software nainstalovat ručně.

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



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*.

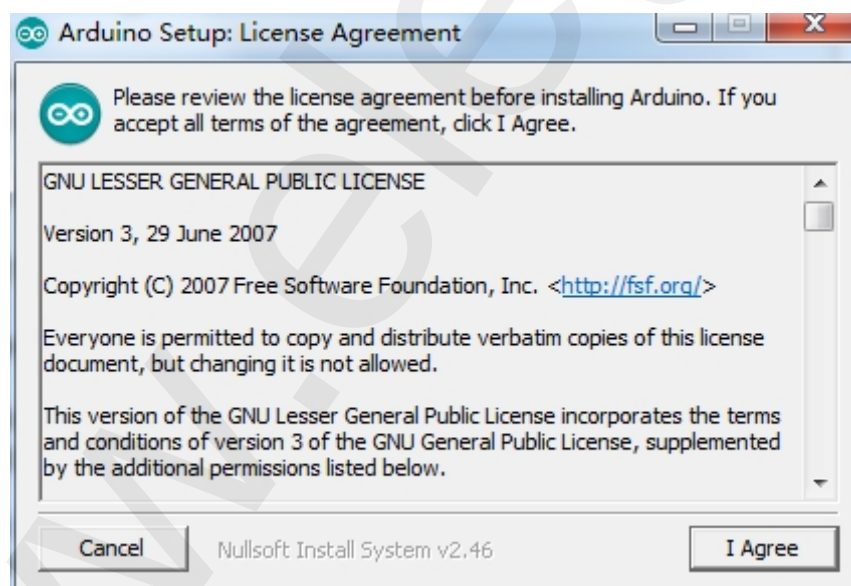
V námi poskytnutých materiálech je k dispozici také verze 1.8.0 a verze našich materiálů jsou nejnovější verze v době, kdy byl tento kurz vytvořen.

 `arduino-1.8.0-linux32.tar.xz`
 `arduino-1.8.0-linux64.tar.xz`
 `arduino-1.8.0-macosx.zip`
 `arduino-1.8.0-windows.exe`
 `arduino-1.8.0-windows.zip`

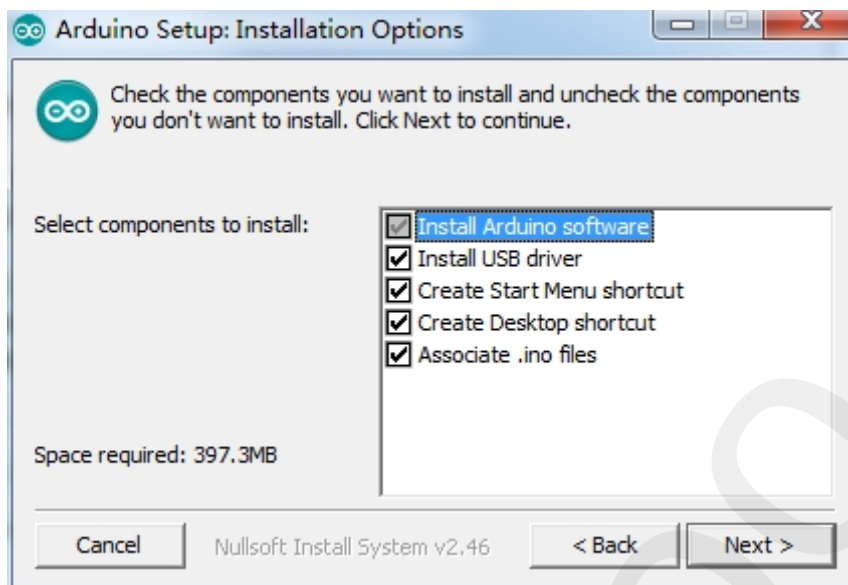
Instalace Arduina (Windows)

Nainstalujte Arduino pomocí exe. Instalační balíček.

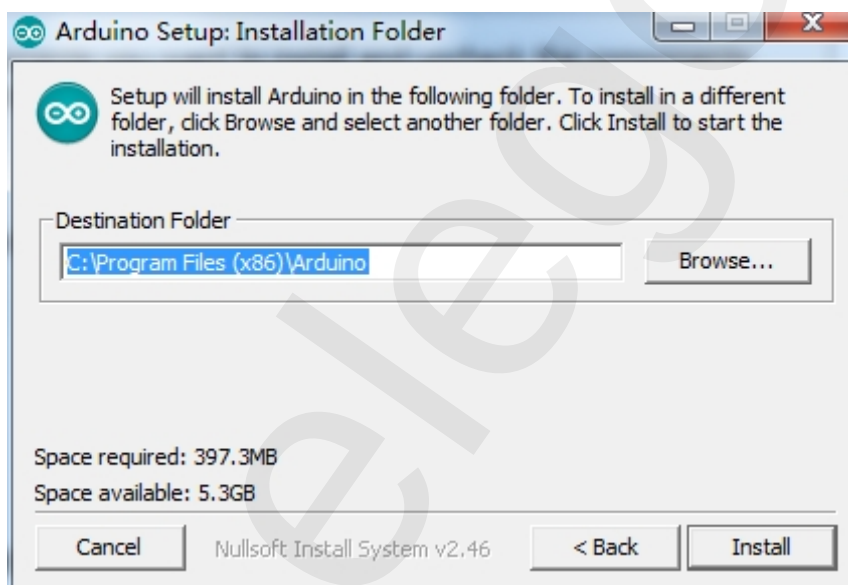
 `arduino-1.8.0-windows.exe`



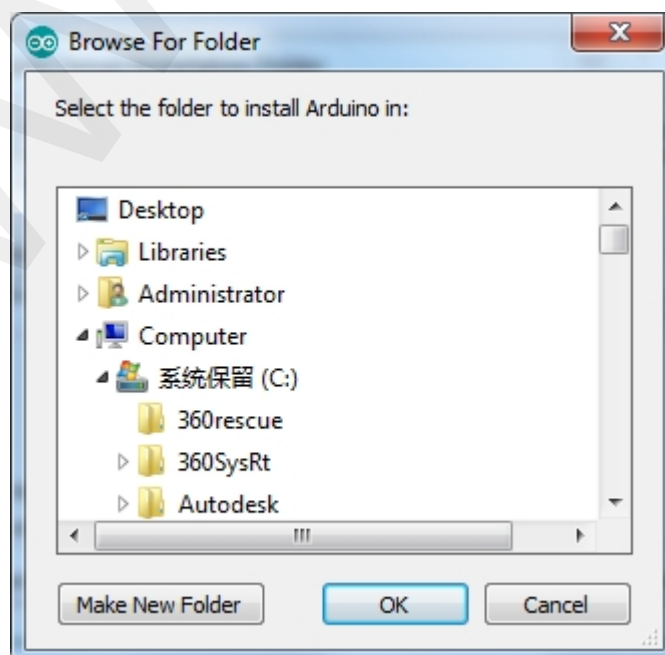
Po kliknutí na tlačítko *Souhlasím* se zobrazí následující rozhraní



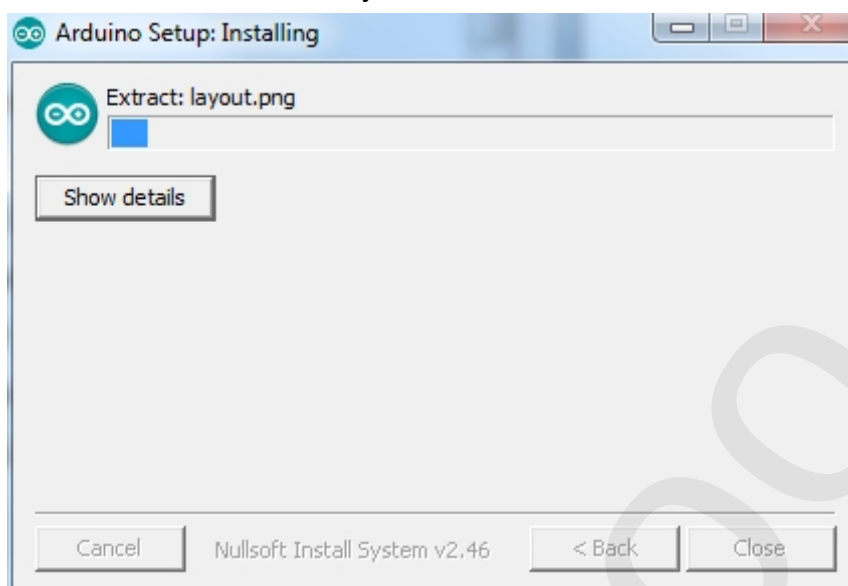
Klikněte na tlačítko *Další*



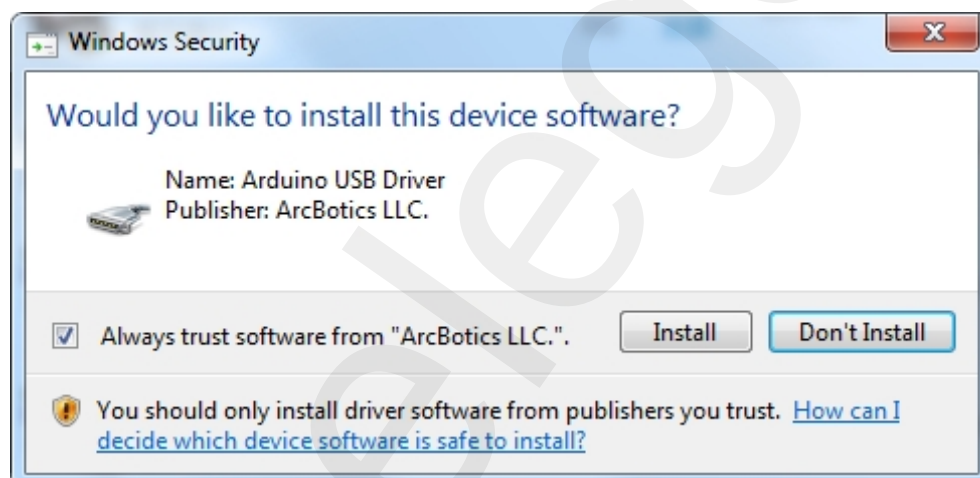
Stisknutím tlačítka **Procházet...** můžete zvolit cestu k instalaci nebo přímo zadat požadovaný adresář.



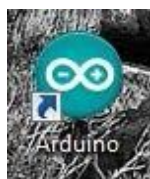
Kliknutím na tlačítko *Instalovat* zahájíte instalaci



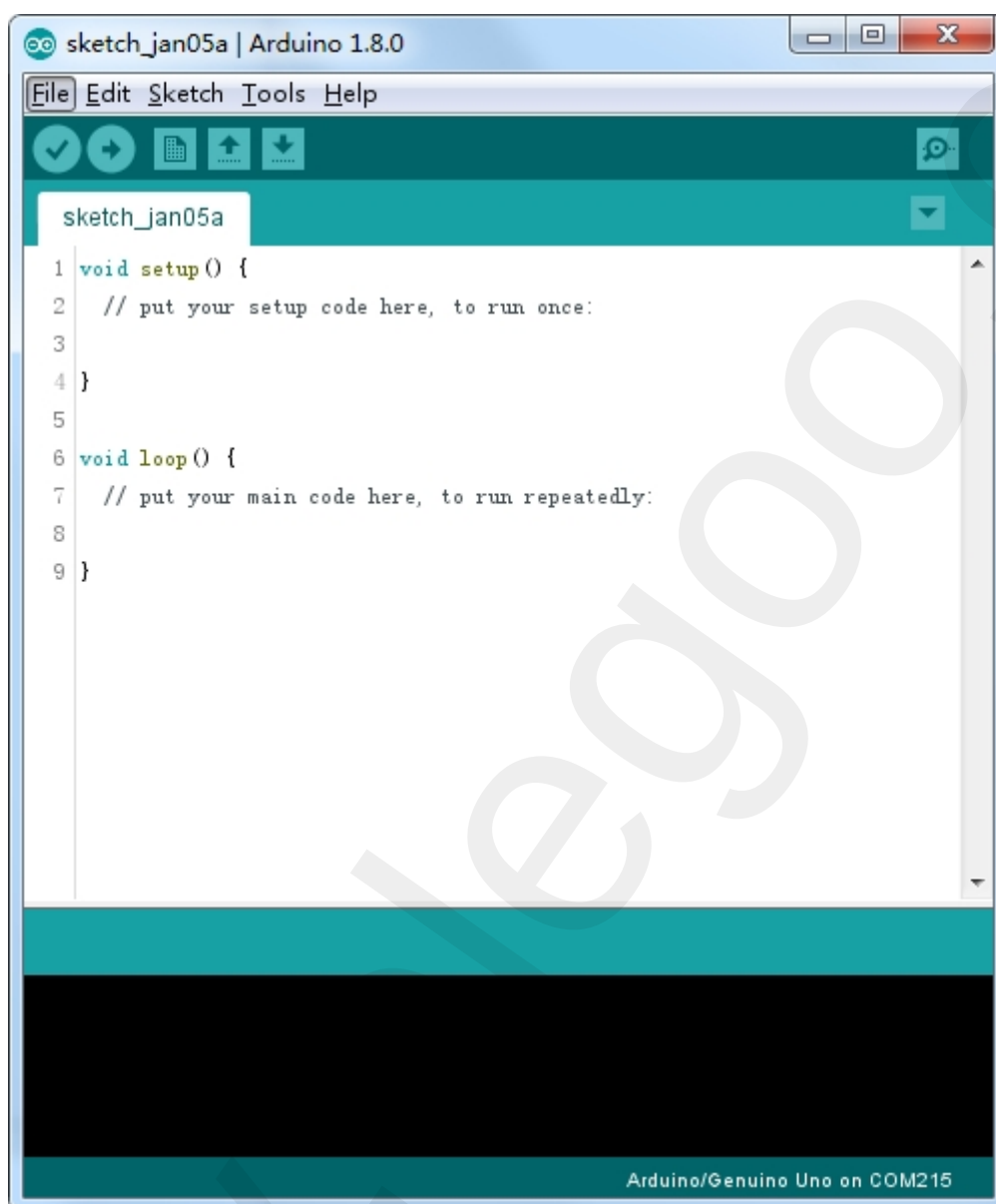
Nakonec se zobrazí následující rozhraní, instalaci dokončete kliknutím na tlačítko *Instalovat*.



Poté se na ploše zobrazí následující ikona



Dvojklikem vstoupíte do požadovaného vývojového prostředí

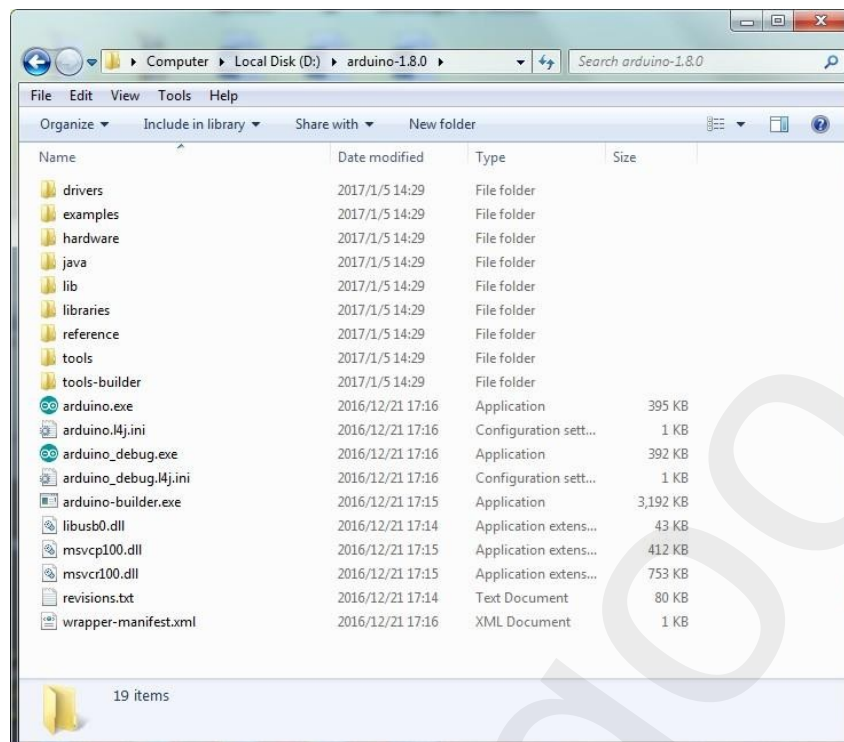


Můžete přímo vybrat instalační balíček pro instalaci a přeskočit obsah níže a přejít na další část. Pokud se však chcete seznámit s jinými metodami než s instalačním balíčkem, pokračujte ve čtení této části.

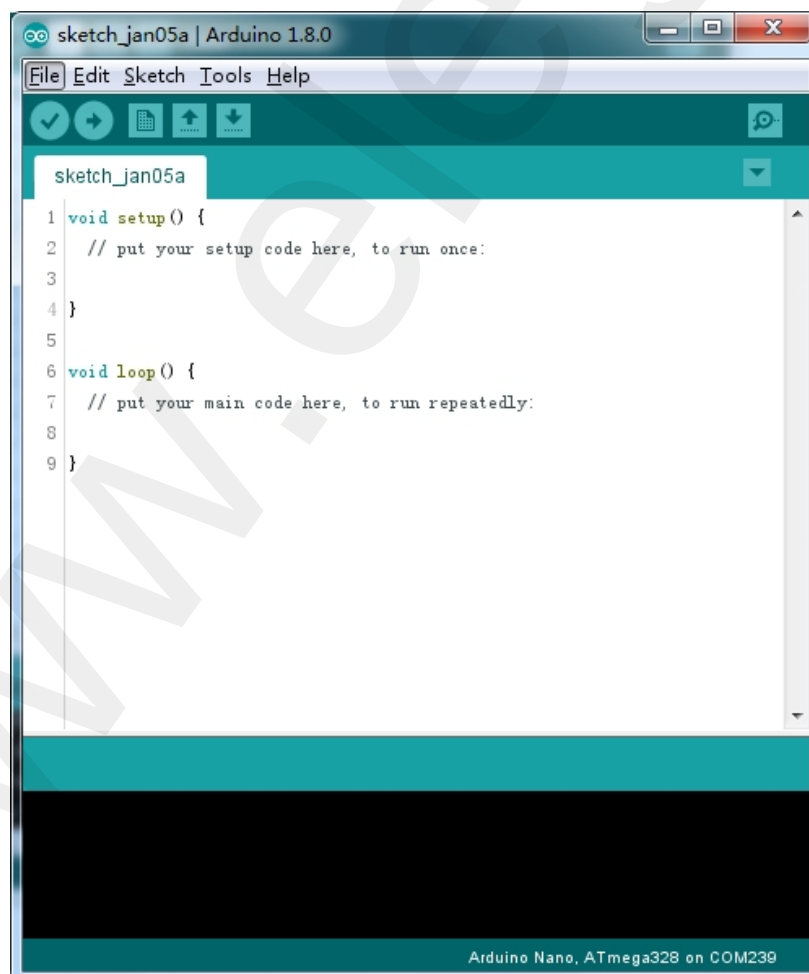
Rozbalte stažený soubor zip, dvojitým kliknutím otevřete program a zadejte požadované vývojové prostředí.



arduino-1.8.0-windows.zip



 **arduino.exe**



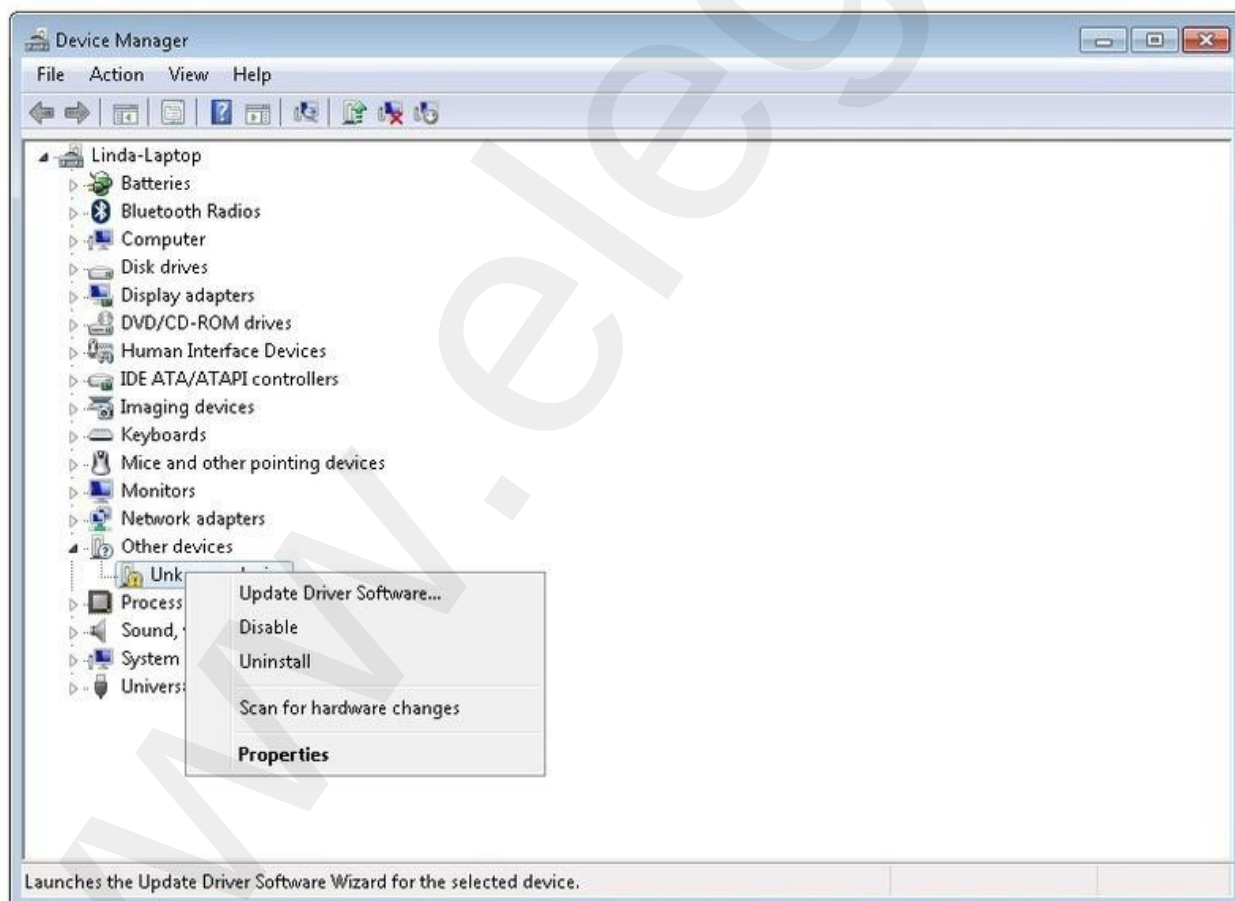
Tento způsob instalace však vyžaduje samostatnou instalaci ovladače.

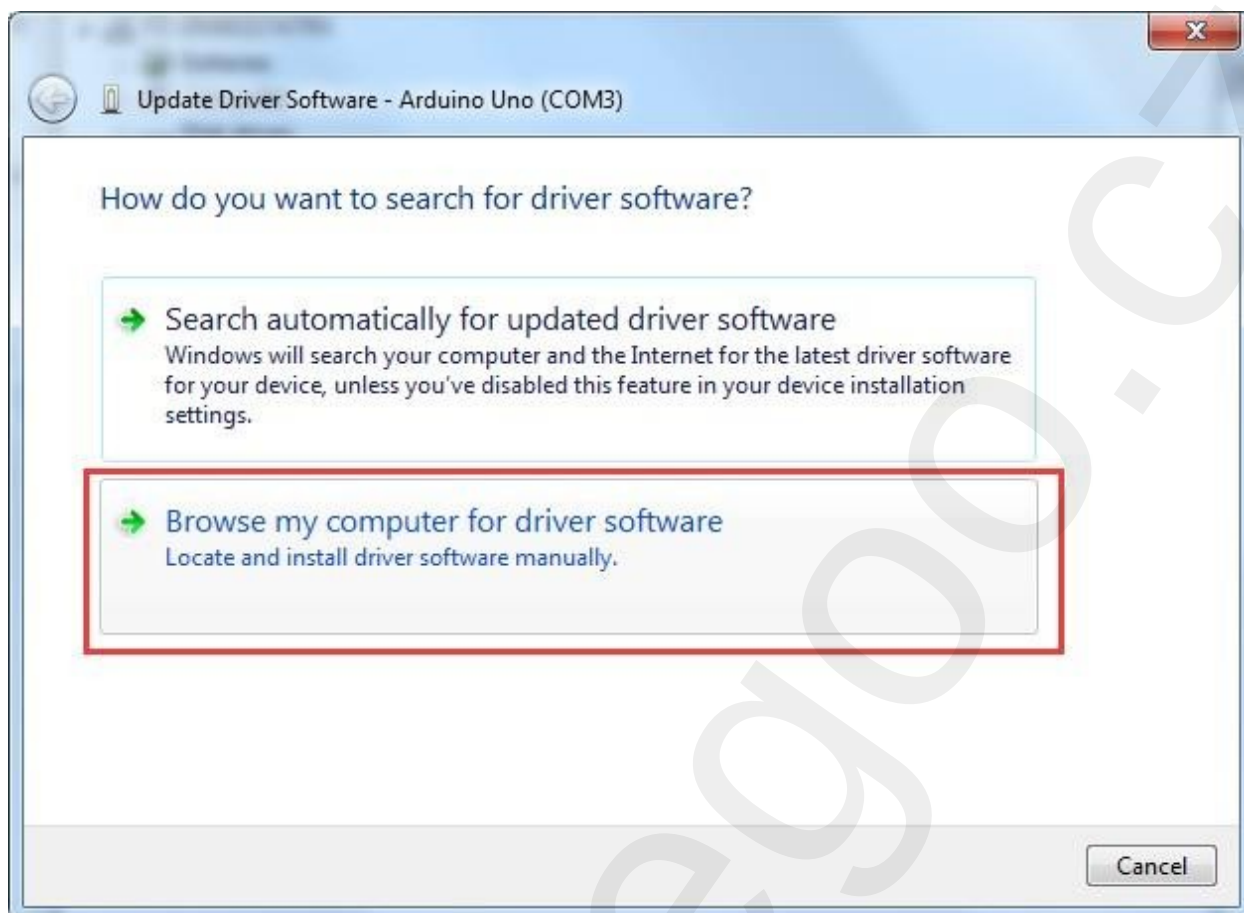
Složka Arduino obsahuje jak samotný program Arduino, tak ovladače, které umožňují připojení Arduina k počítači pomocí kabelu USB. Než spustíme program Arduino, nainstalujete ovladače USB.

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.

www.elegoo.cz

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)

Stáhněte a rozbalte soubor ZIP, poklepejte 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-1.8.0-macosx.zip

Instalace Arduina (Linux)

Musíte použít příkaz `make install`. Pokud používáte systém Ubuntu, doporučujeme nainstalovat Arduino IDE ze softwarového centra Ubuntu.

 `arduino-1.8.0-linux32.tar.xz`

 `arduino-1.8.0-linux64.tar.xz`

TIPY: Pokud máte problémy s instalací ovladačů, přečtěte si prosím UNO R3, MEGA, NANO DRIVER FAQ.

 `UNO R3, MEGA, NANO DRIVER FAQ`

Lekce 1 Přidání knihoven a otevření sériového monitoru

Instalace dalších knihoven Arduina

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

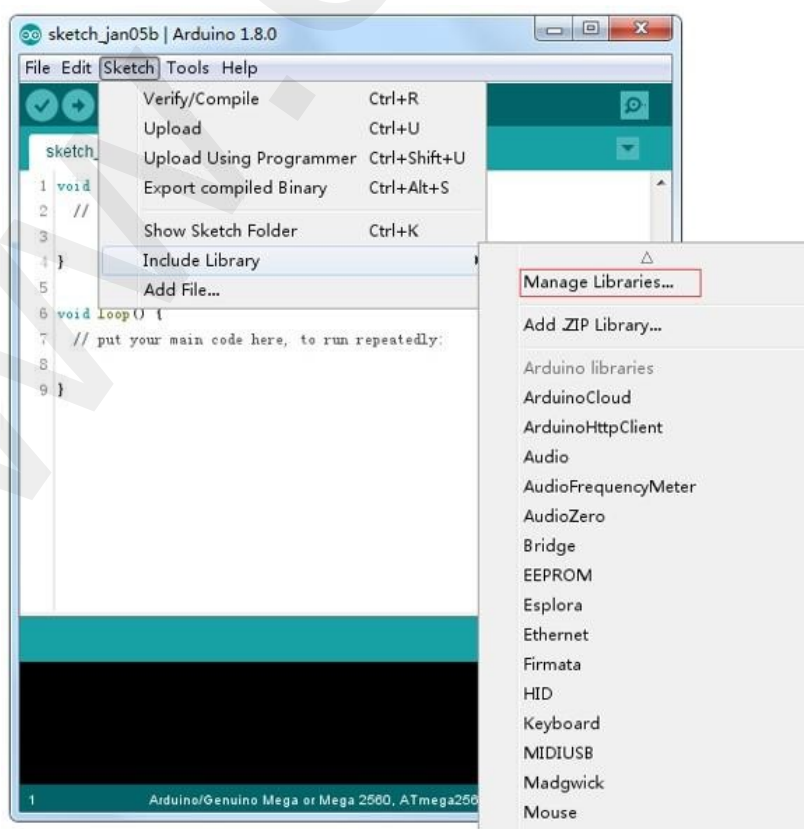
Co jsou knihovny?

Knihovny představují soubor kódu, který usnadňuje připojení k senzoru, displeji, modulu atd. Například vestavěná knihovna LiquidCrystal 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

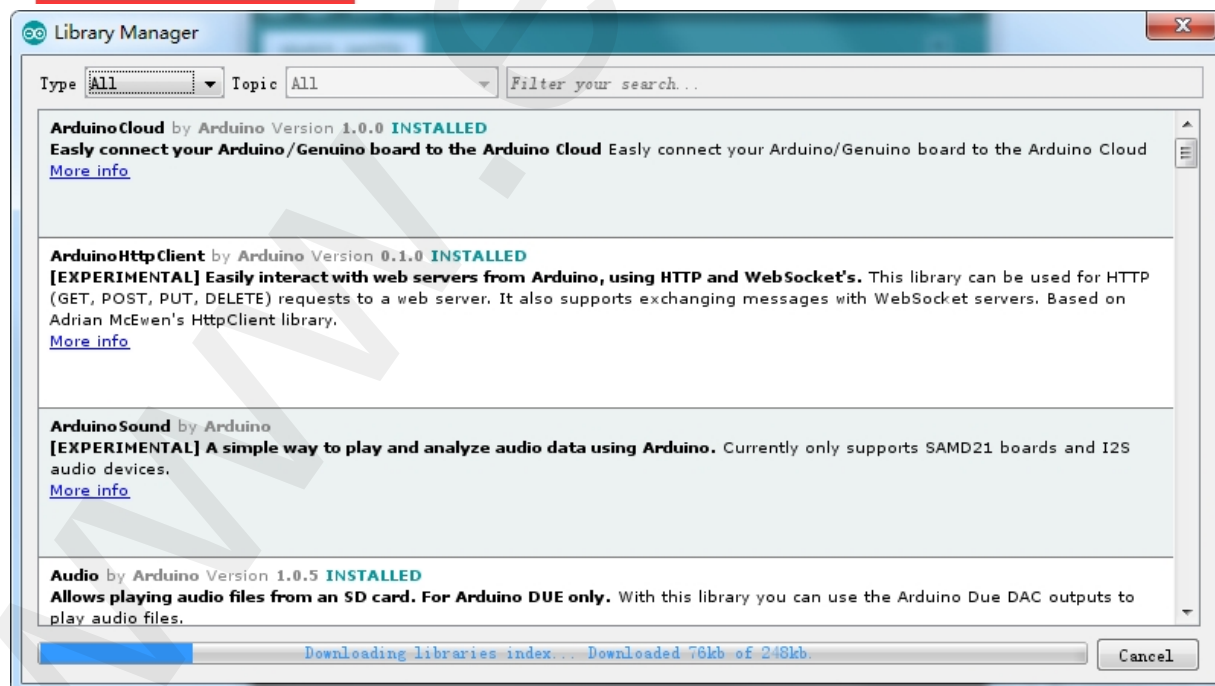
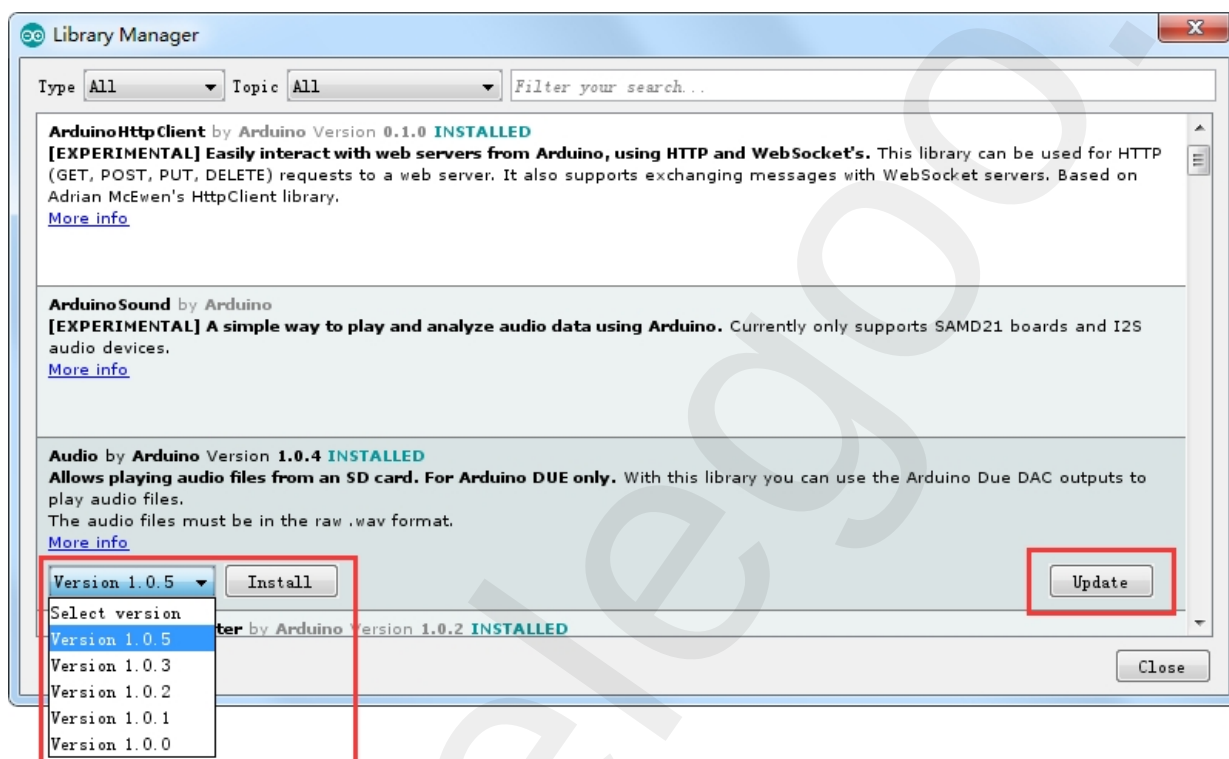
Používání Správce knihovny

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.

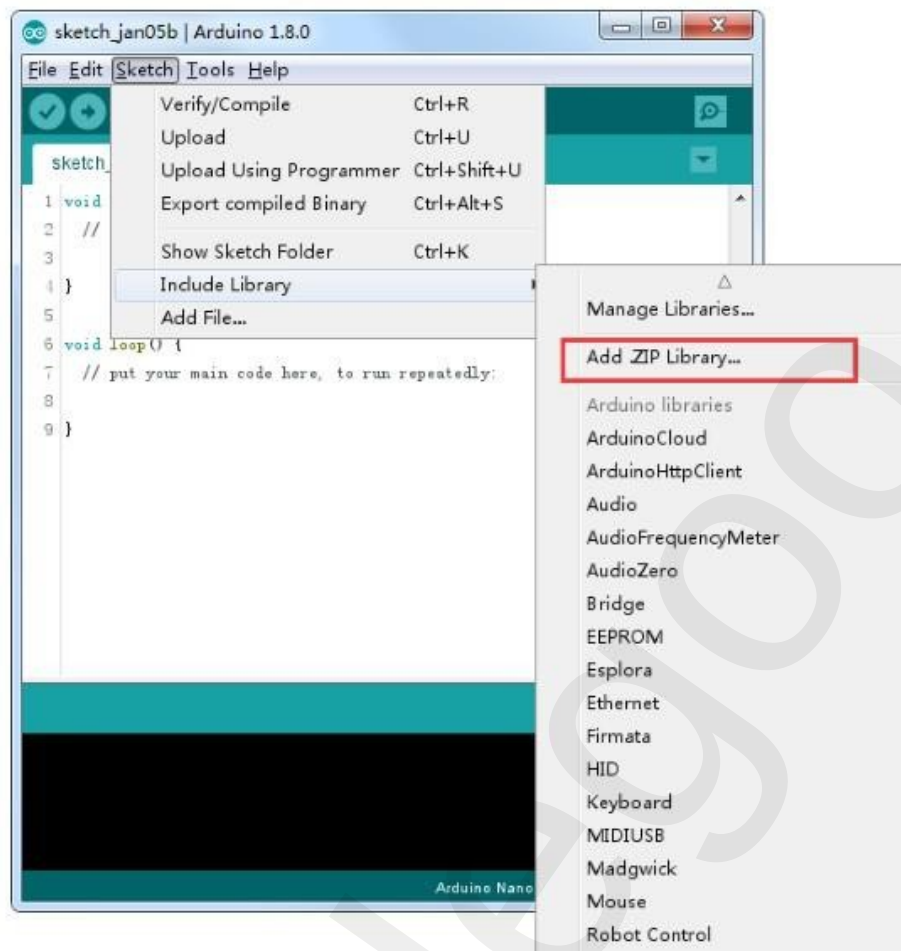


Novou knihovnu nyní najdete v nabídce Zahrnout knihovnu. Pokud chcete přidat vlastní knihovnu, otevřete nový problém na [Githubu](#).

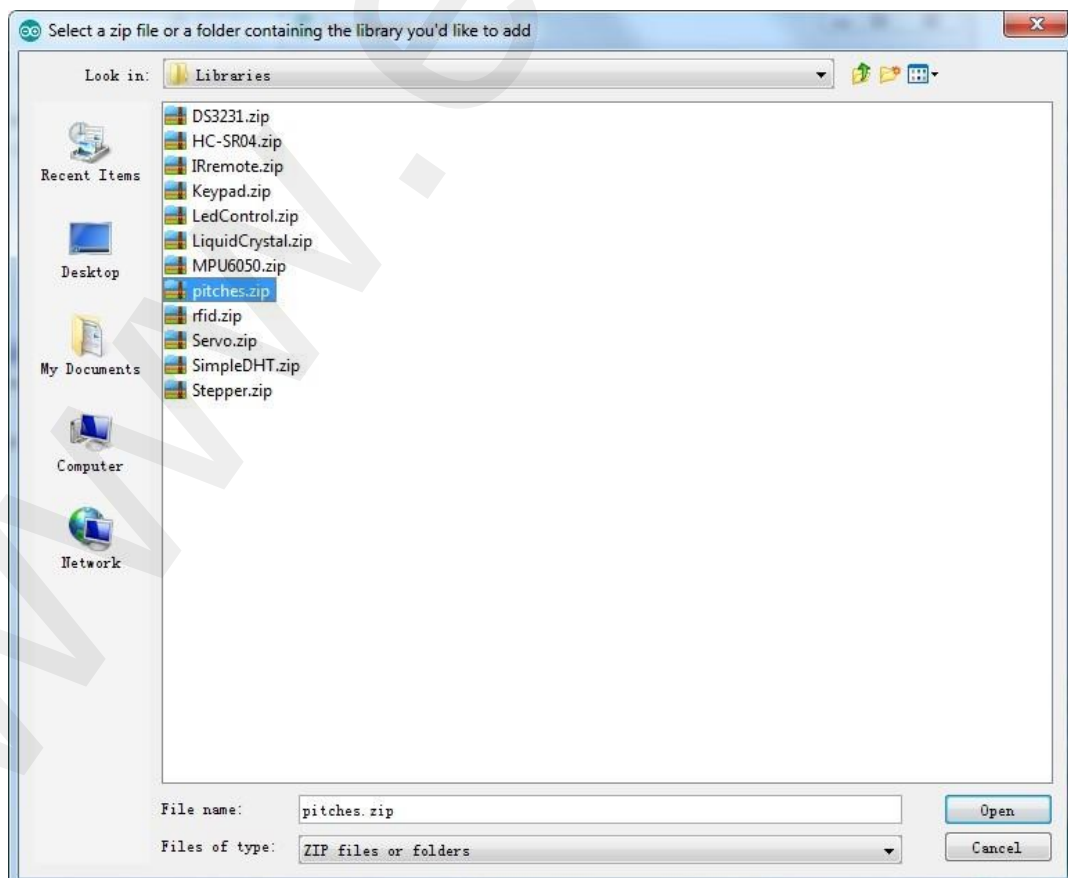
Importování knihovny .zip

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 rozevíracího 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. Knihovna by se nyní měla zobrazit v dolní části rozbalovací nabídky. Je připravena k použití ve vašem náčrtu. Soubor ZIP bude rozbalen ve složce libraries v adresáři Arduino sketches. **Poznámka: knihovna bude k dispozici pro použití ve skicích, 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 zřídka a uživatelé bez potřeb ji mohou přeskóčit.

Ruční instalace

Chcete-li knihovnu nainstalovat, nejprve ukončete aplikaci Arduino. Poté rozbalte soubor ZIP obsahující knihovnu. Pokud například instalujete knihovnu s názvem

"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 ArduinoParty. Pod Windows se bude pravděpodobně jmenovat "Moje dokumenty\Arduino\libraries". Pro uživatele Macu se bude pravděpodobně jmenovat "Documents/Arduino/libraries". V Linuxu to bude složka "libraries" ve vašem skicáku.

Vaše složka s knihovnou Arduino by nyní měla vypadat takto (ve

Windows): **Moje**

dokumenty\Arduino\knihovny\ArduinoParty\ArduinoParty.cpp

Moje

dokumenty\Arduino\knihovny\ArduinoParty\ArduinoParty.h

Moje dokumenty\Arduino\knihovny\ArduinoParty\příklady

nebo takto (v systémech Mac a Linux):

Dokumenty/Arduino/knihovny/ArduinoParty/ArduinoParty.cpp

Dokumenty/Arduino/knihovny/ArduinoParty/ArduinoParty.h

Dokumenty/Arduino/knihovny/ArduinoParty/příklady

....

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!

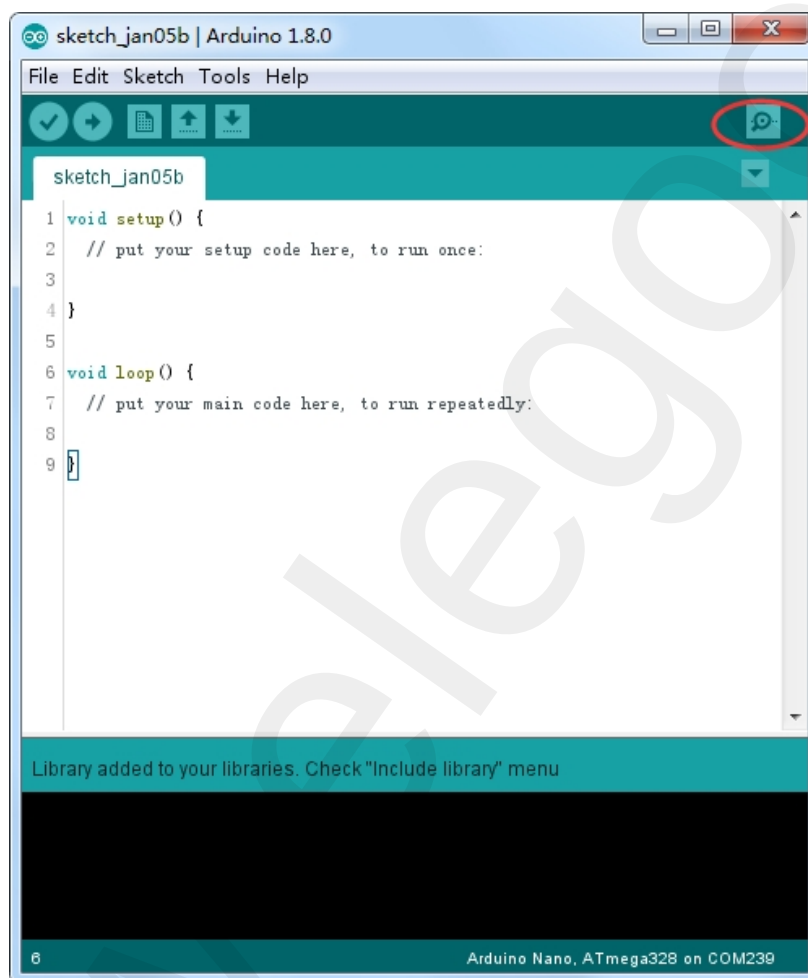
Arduino Serial Monitor (Windows, Mac, Linux)

Integrované vývojové prostředí (IDE) Arduino je softwarová část platformy Arduino. A protože používání terminálu je tak důležitou součástí práce se systémem

Arduina a dalších mikrokontrolérů, rozhodli se k softwaru připojit sériový terminál. V prostředí Arduino se tento terminál nazývá Serial Monitor.

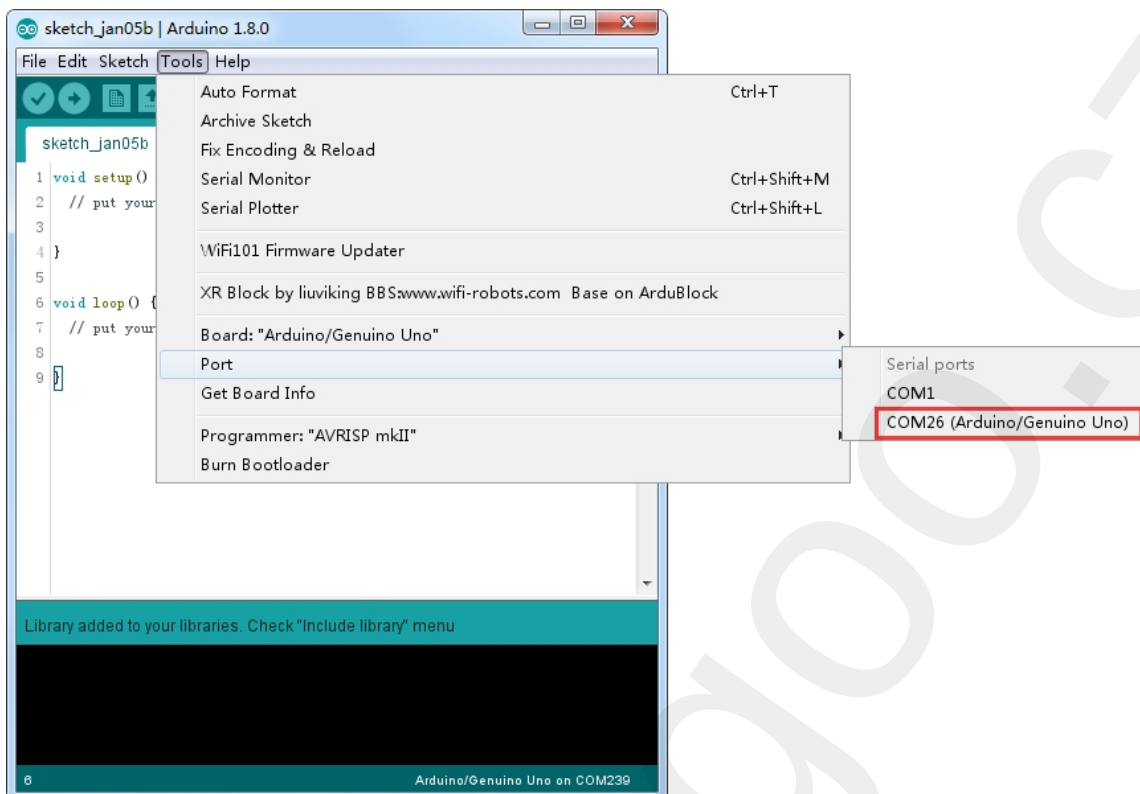
Navázání spojení

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

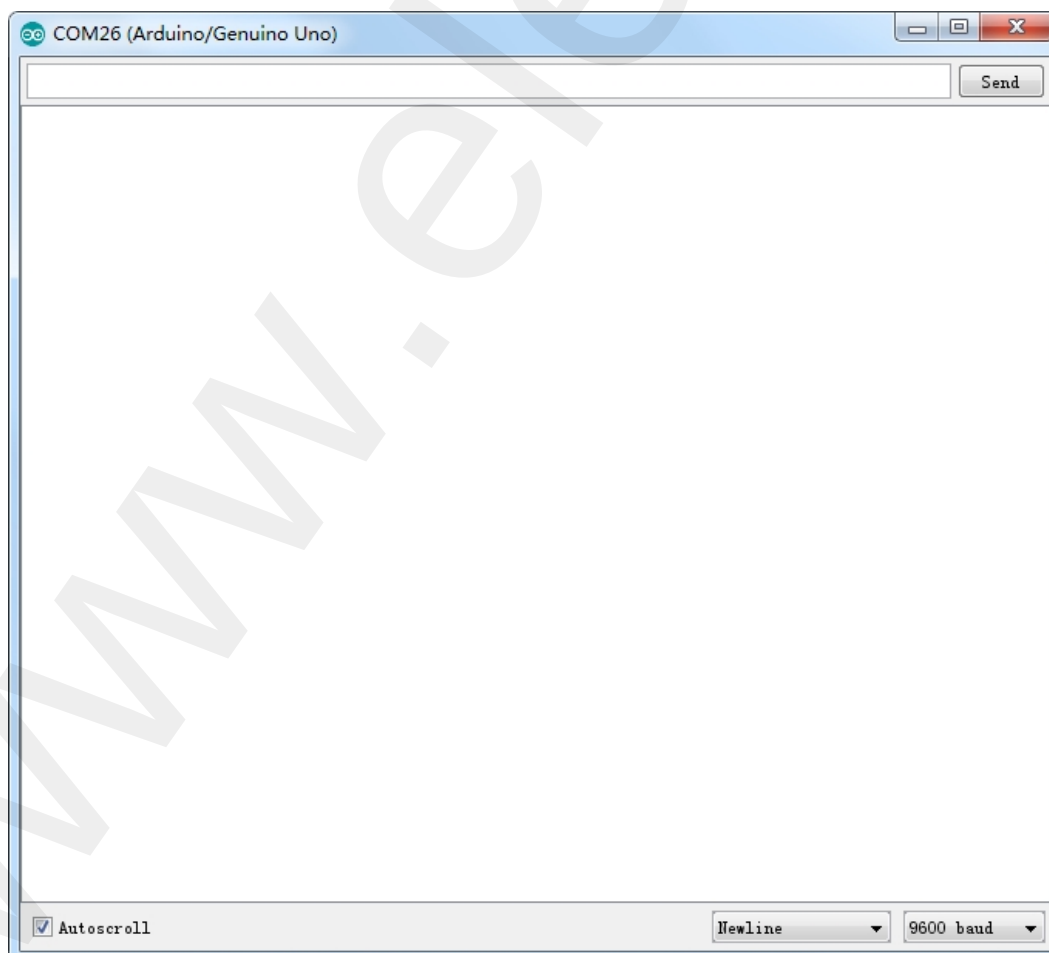


Výběr portu, který se má otevřít v Sériovém monitoru, je stejný jako výběr portu pro nahrávání kódu Arduina. Přejděte na Tools -> Serial Port a vyberte správný port.

Tipy: Zvolte stejný port COM, který máte ve Správci zařízení.

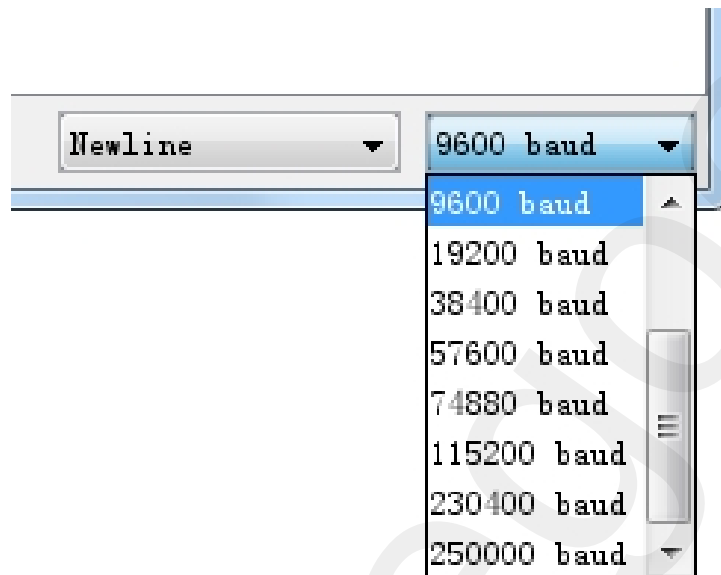


Po otevření byste měli vidět něco podobného:

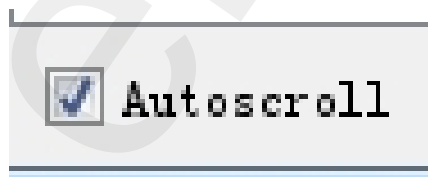


Nastavení

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, zda se má terminál automaticky posouvat, nebo ne.



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 Blink

Přehled

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

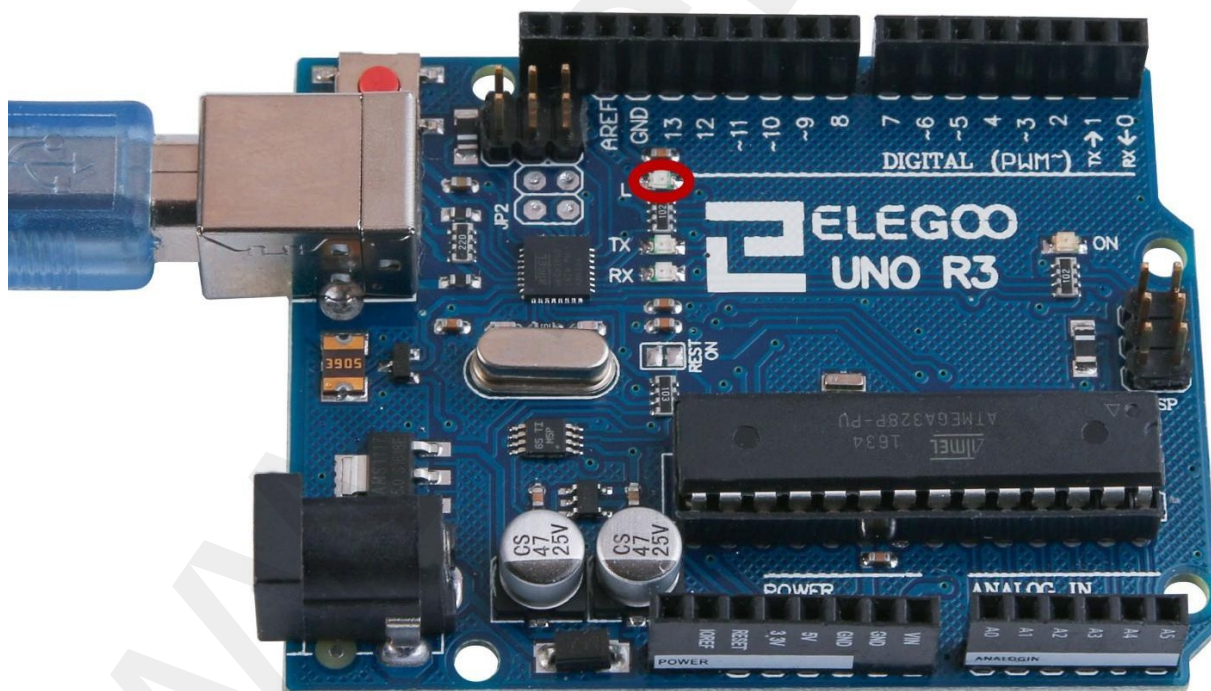
Požadovaná součást:

(1) x Elegoo Uno R3

Princip

Deska UNO R3 má po obou stranách řadu konektorů, které slouží k připojení k několika elektronickým zařízením, a zásuvné "štíty", které rozšiřují její možnosti.

Má také jednu LED diodu, kterou můžete ovládat ze svých náčrtů. Tato LED dioda je zabudovaná na desce UNO R3 a často se označuje jako LED "L", protože je tak na desce označena.



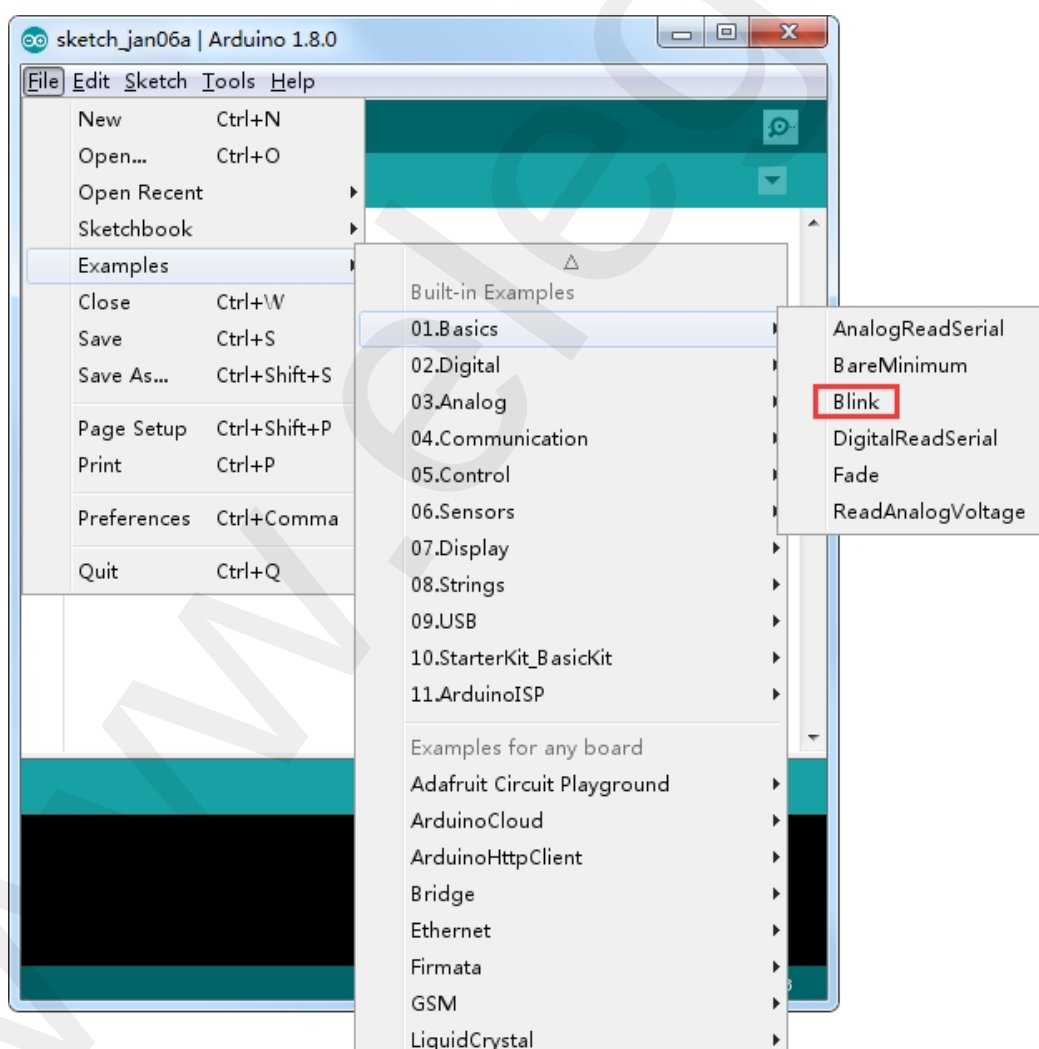
Možná zjistíte, že LED dioda "L" na desce UNO R3 bliká již při připojení do zástrčky USB. Je to proto, že desky jsou obvykle dodávány s předinstalovaným skečem 'Blink'.

V této lekci přeprogramujeme desku UNO R3 vlastním náčrtem blikání a poté změníme rychlost blikání.

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

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

Nahrajte sketch 'Blink', který najdete v nabídce IDE pod položkou File > Examples > 01.Basics.



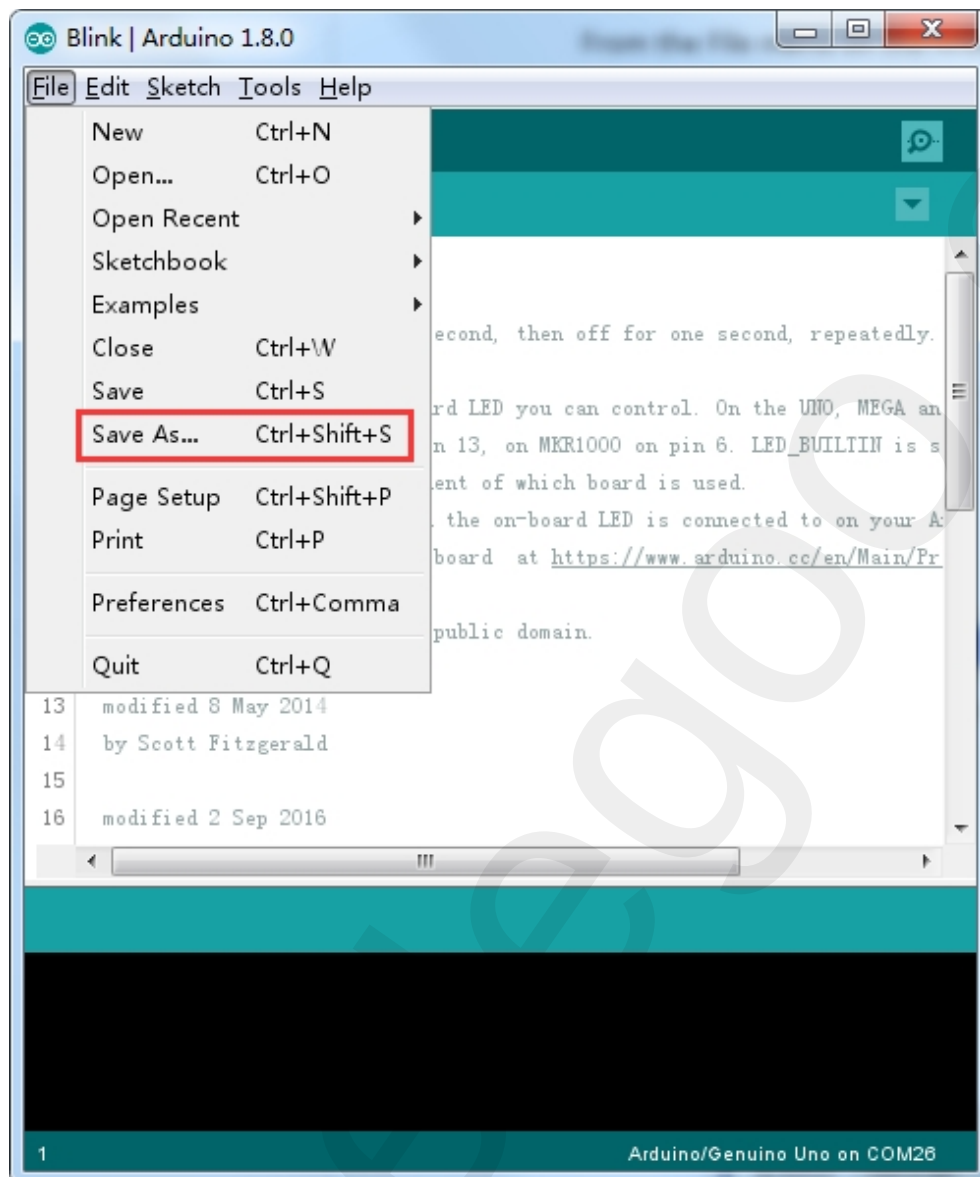
Po otevření okna náčrtu jej zvětšete tak, abyste v něm viděli celý náčrt.

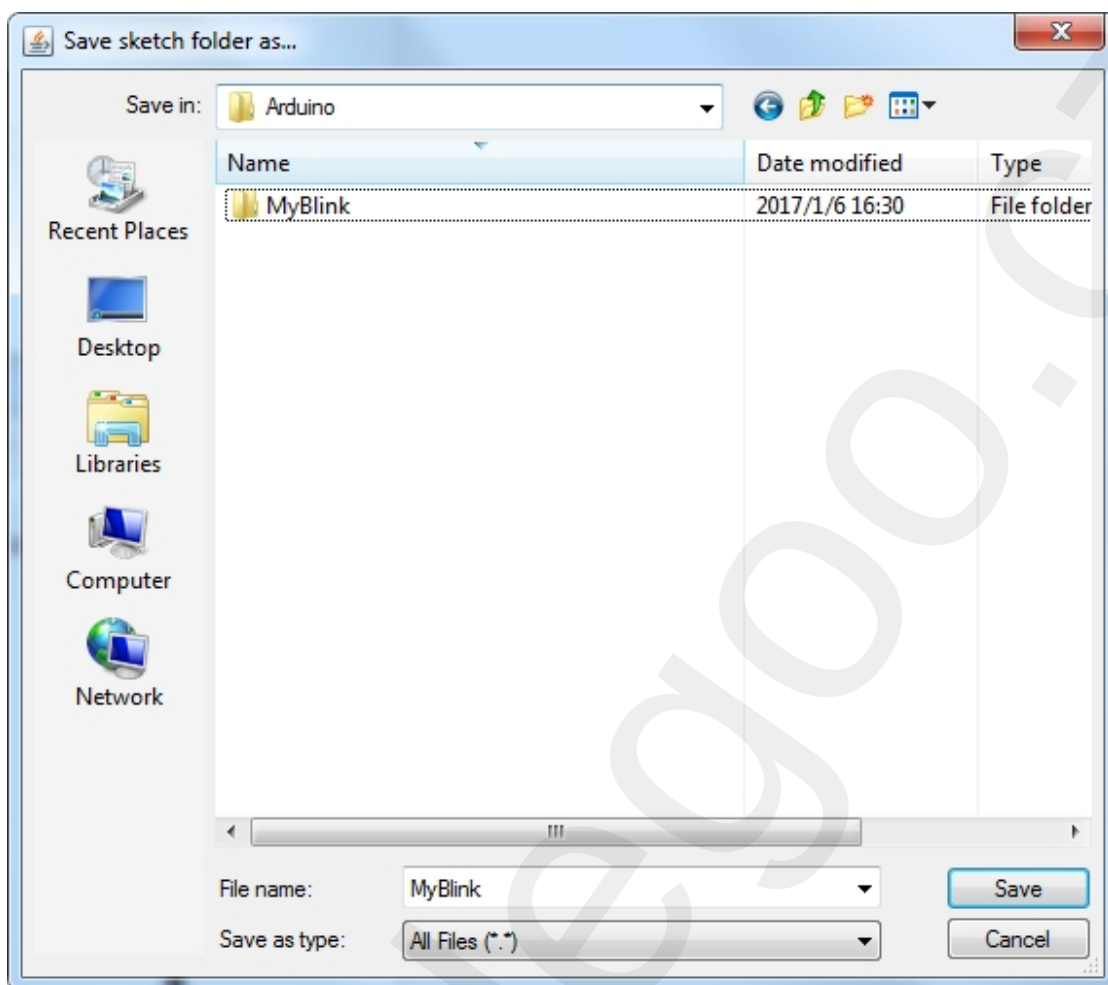


Příkladové náčrty, které jsou součástí Arduino IDE, jsou určeny pouze pro čtení. To znamená, že je můžete nahrát na desku UNO R3, ale pokud je změníte, nemůžete je uložit jako stejný soubor.

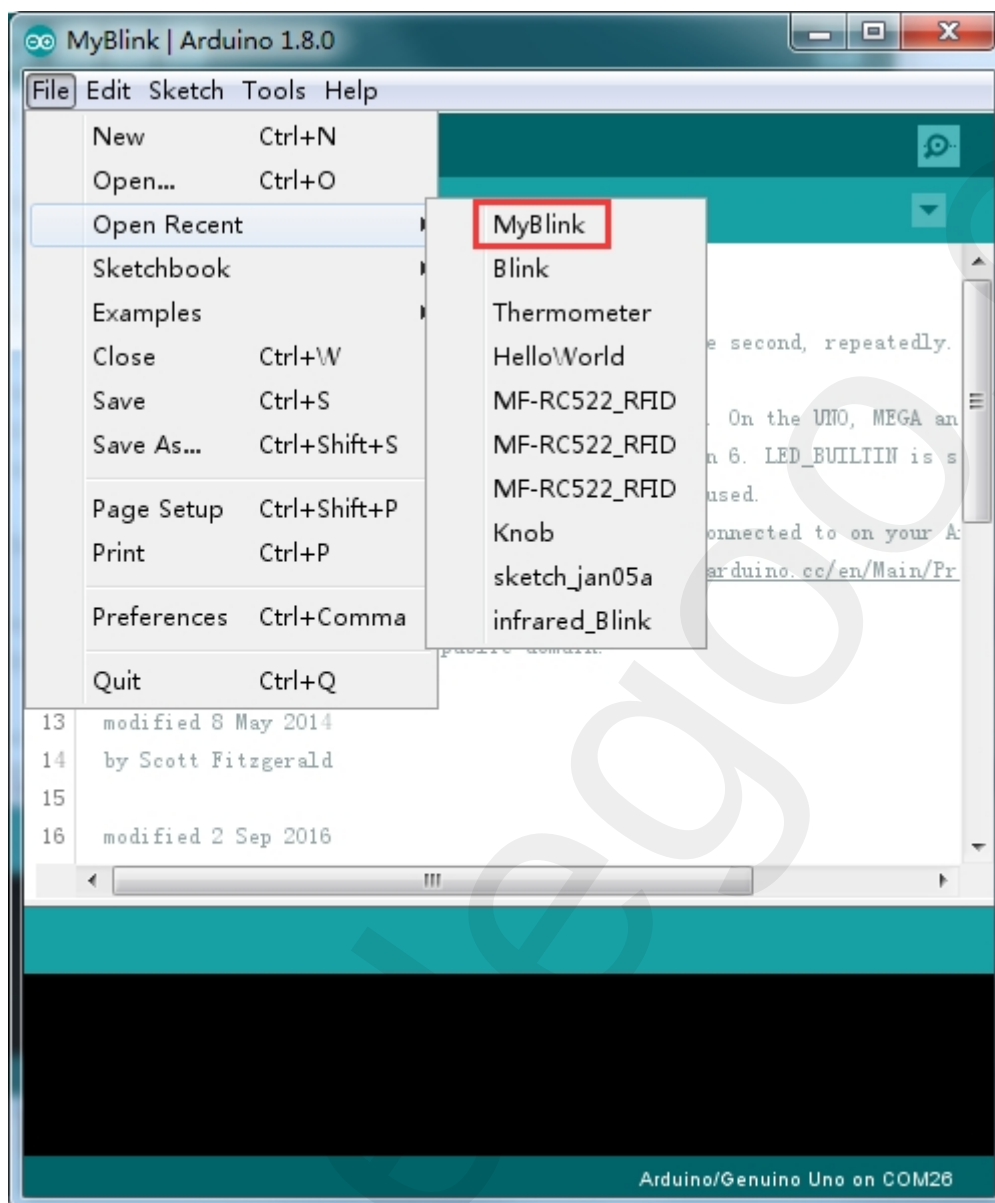
Protože se chystáme tento náčrt změnit, musíte nejprve uložit svou vlastní kopii.

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

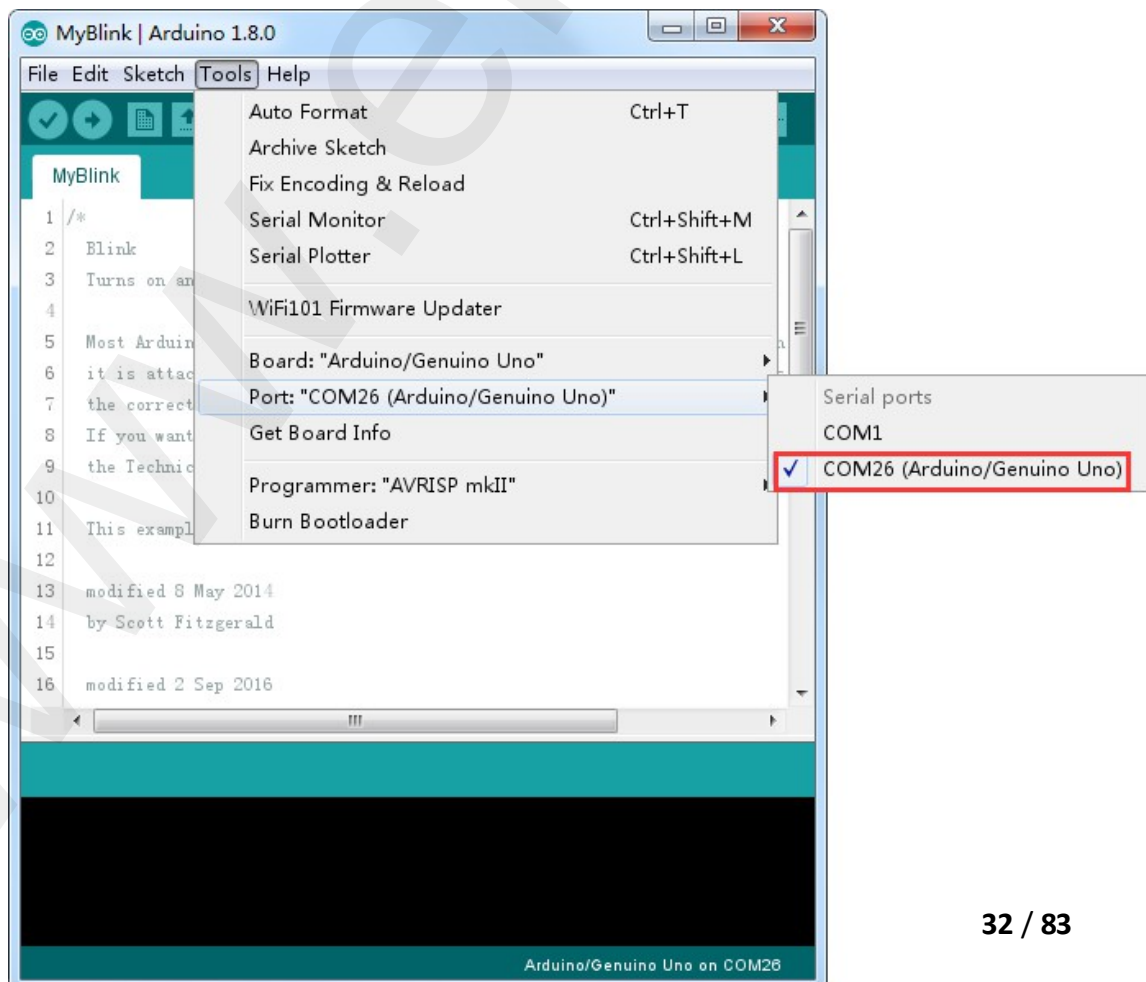
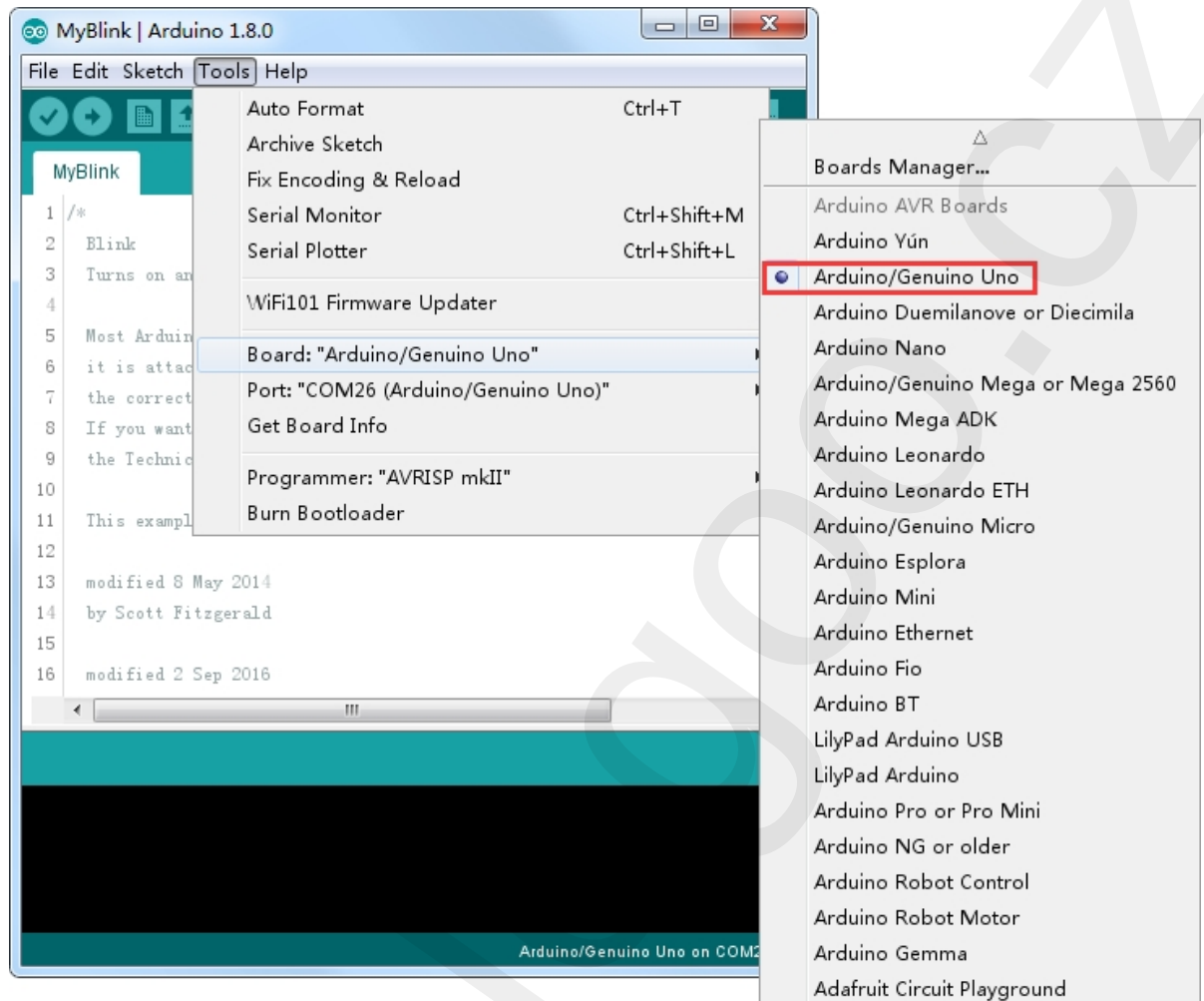




Uložili jste si do skicáku kopii obrázku "Blink". To znamená, že pokud ji budete chtít znovu najít, stačí ji otevřít pomocí nabídky Soubor > Skicák.

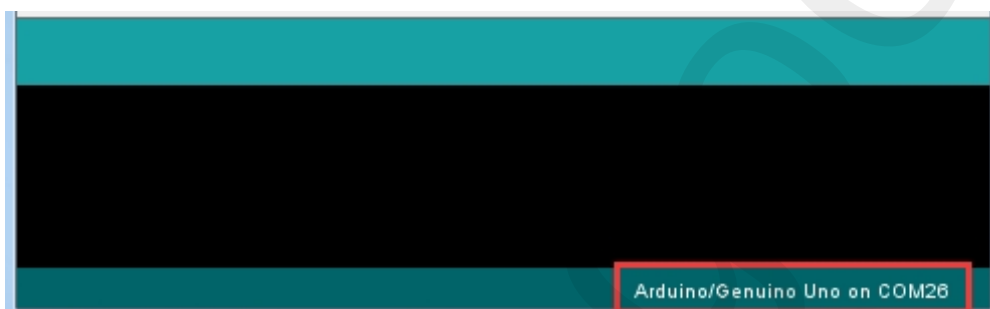


Připojte desku Arduino k počítači pomocí kabelu USB a zkontrolujte, zda jsou správně nastaveny položky "Board Type" a "Serial Port".



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 26, může to být COM3 nebo COM4 ve vašem počítači. Správný COM port má být COMX (arduino XXX), což je podle certifikačních kritérií.

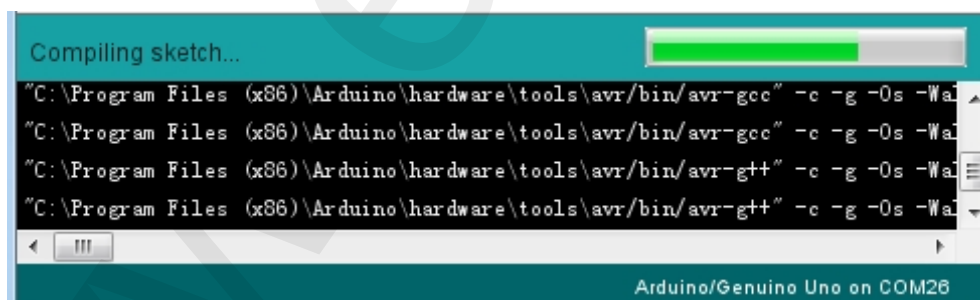
Arduino IDE vám v dolní části okna zobrazí aktuální nastavení desky.



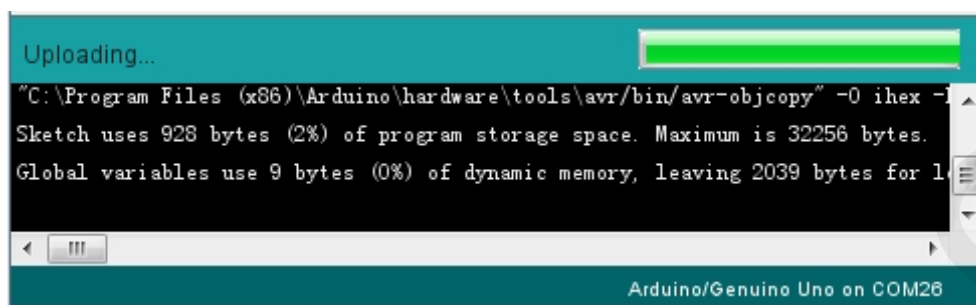
Klikněte na tlačítko "Nahrát". Druhé tlačítko zleva na panelu nástrojů.



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 náčrtek využívá 928 bajtů z dostupných 32 256 bajtů. Po fázi "Compiling Sketch.." se může zobrazit následující chybová zpráva:



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 lekcí 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. U většiny Arduin včetně UNO a Leonarda je to 13.

Dále máme funkci "setup". Jak je uvedeno v komentáři, tato funkce se opět spustí 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, LOW);  //vypnutí LED diody pomocí napětí LOW  
  delay(1000);             // čekání po dobu jedné sekundy  
}
```

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 přimějte 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 blikat rychleji.

Lekce 3 LED

Přehled

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

Požadovaná součást:

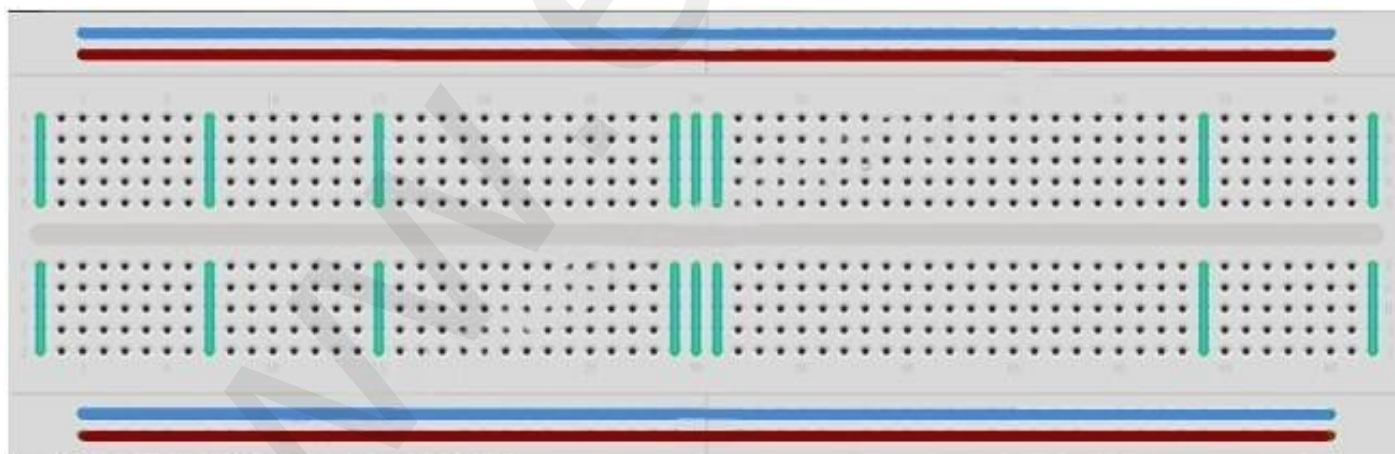
- (1) x Elegoo Uno R3
- (1) x 5 mm červená LED dioda
- (1) x 220 ohmů odpor
- (1) x 1k ohmový odpor
- (1) x 10k ohmový odpor
- (2) x vodiče M-M (propojovací vodiče typu Male to Male)

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 spojením 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 špatným směrem a 2) LED dioda musí být použita s rezistorem, který omezí nebo "přiškrtní" množství proudu, který jí protéká; jinak shoří!



Pokud u LED diody nepoužijete rezistor, může dojít k jejímu téměř okamžitému zničení, protože jí protéká příliš velký proud, který ji zahřívá a ničí "spoj", na němž vzniká světlo.

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

Za prvé, kladný vodič je delší.

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.

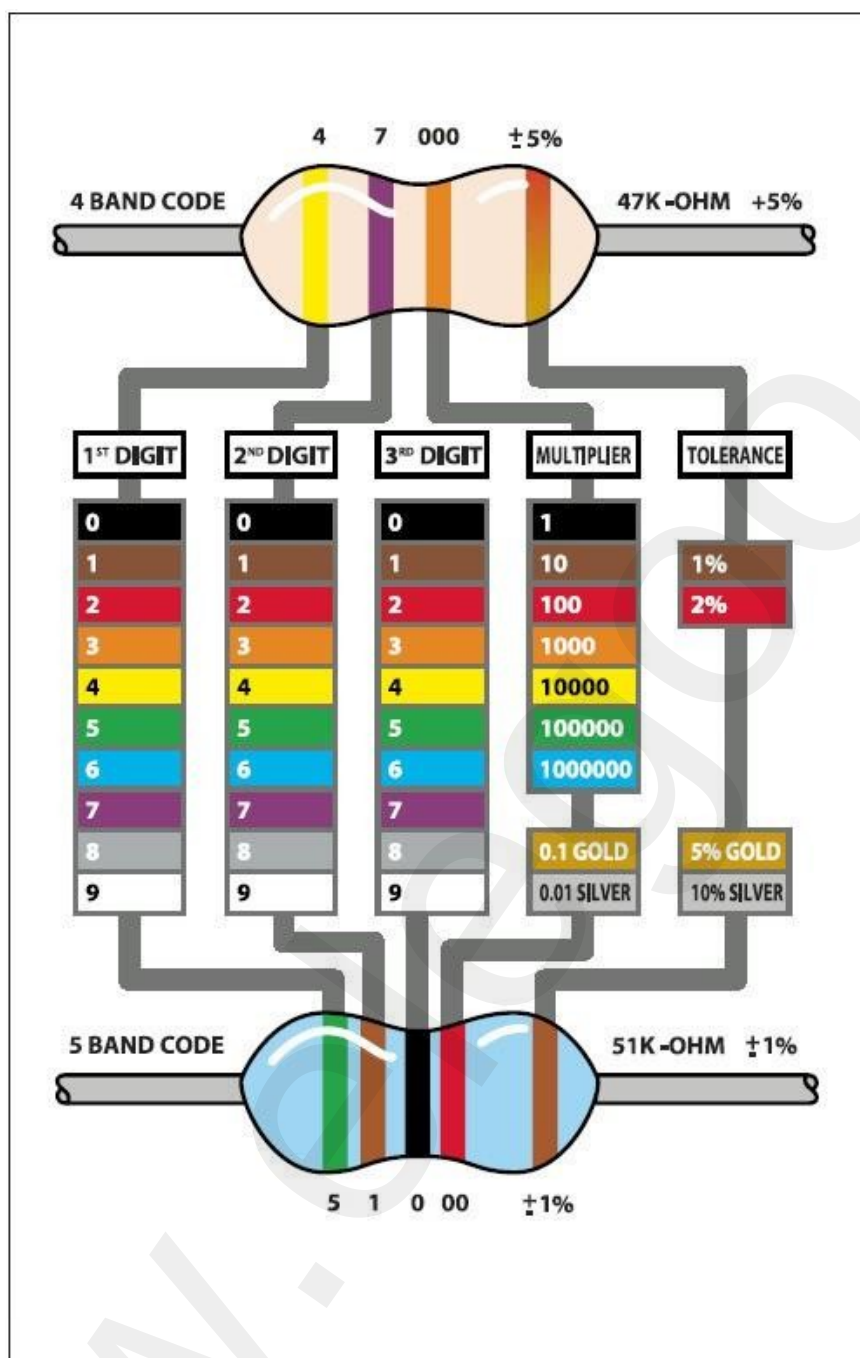


Nejprve však více o rezistorech...

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: 220 Ω , 1k Ω a 10k Ω . 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

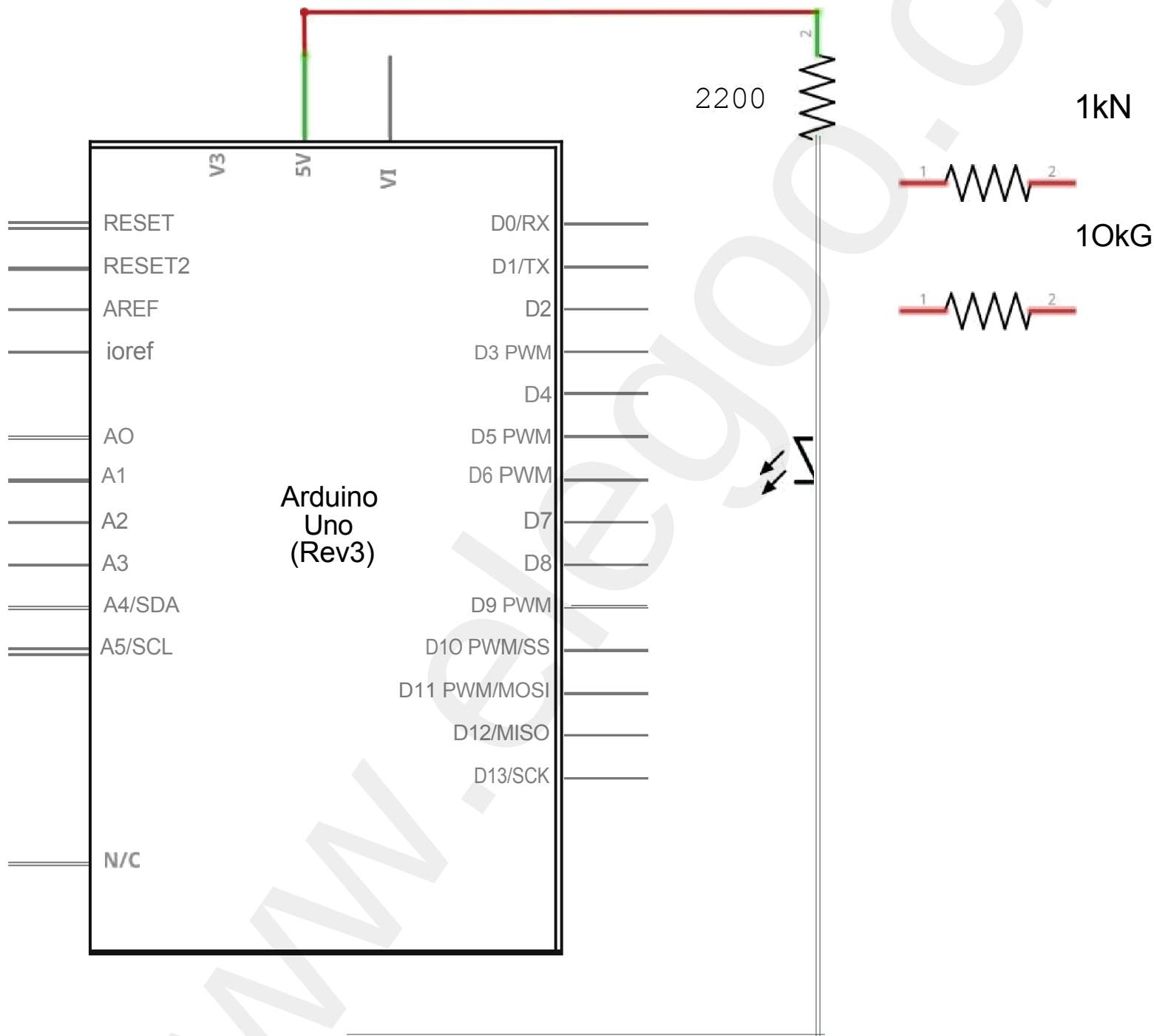
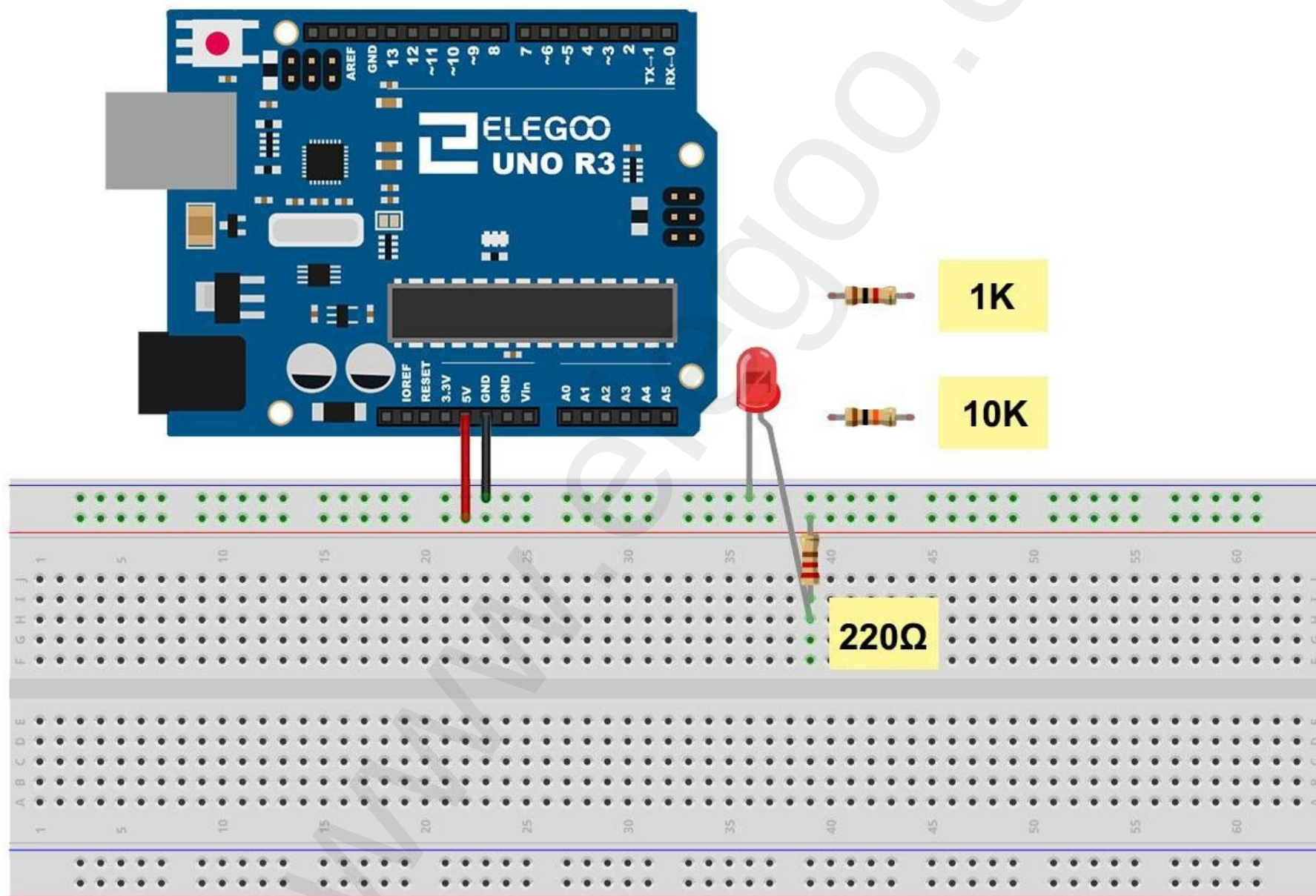


Schéma zapojení



UNO je vhodným zdrojem 5 V, které použijeme k napájení LED a rezistoru. S UNO nemusíte dělat nic jiného, než ho připojit ke kabelu USB.

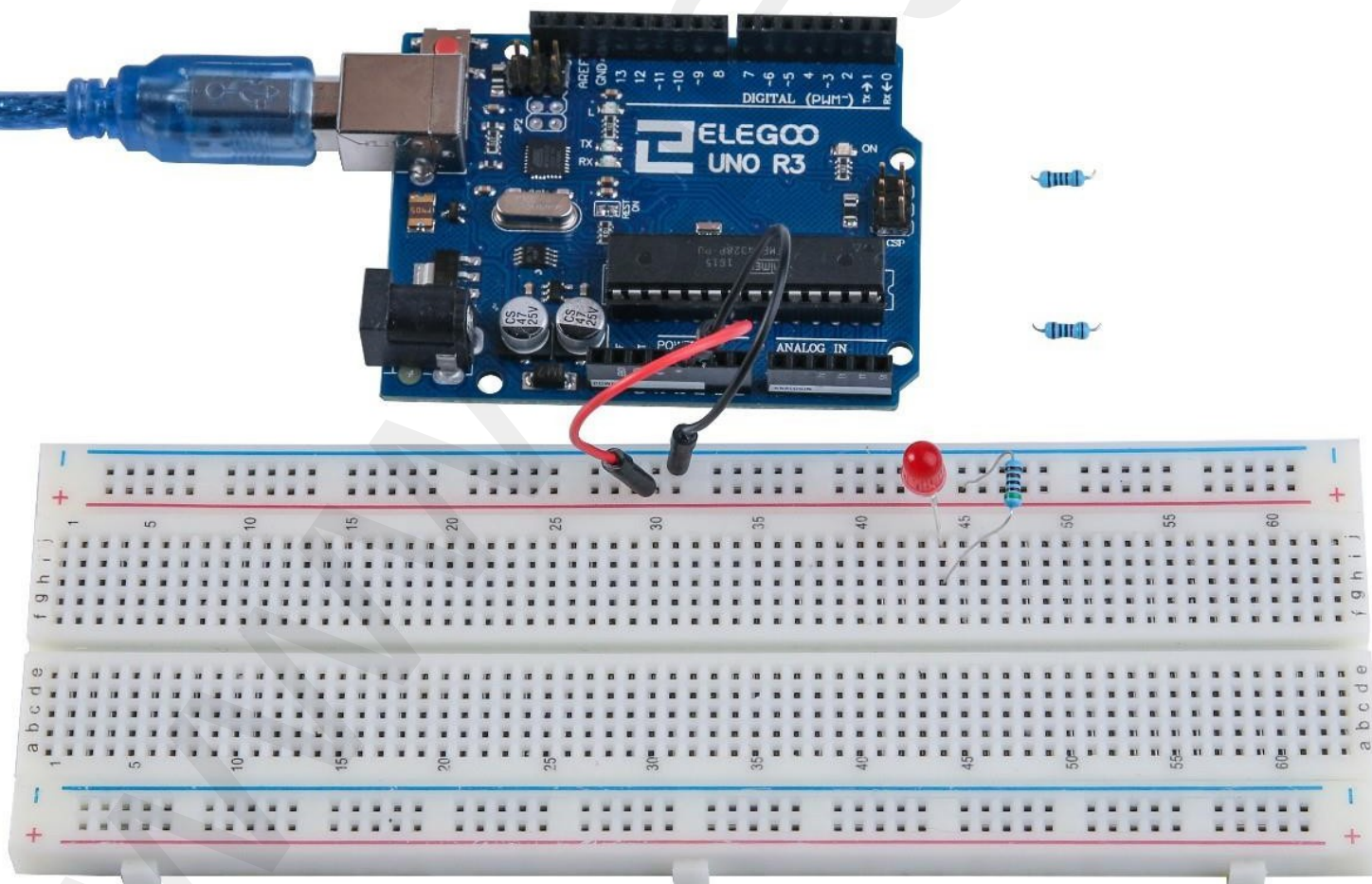
S rezistorem 220 Ω na místě by měla být LED dioda poměrně jasná. Pokud vyměníte rezistor 220 Ω za rezistor 1 k Ω , bude LED dioda o něco slabší. A konečně, s rezistorem 10 k Ω na místě bude LED dioda téměř viditelná. Vytáhněte červený vodič propojky z chlebníku a dotkněte se ho v otvoru a vyjměte ho,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

V tuto chvíli 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 obrázku



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: společná anoda nebo společná katoda. Společná anoda využívá 5 V na společném pinu, zatímco společná katoda se připojuje k zemi.

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 Uno R3
- (1) x 830 vazebních bodů Breadboard
- (4) x vodiče M-M (propojovací vodiče typu Male to Male)
- (1) x RGB LED
- (3) x 220 ohmů rezistory

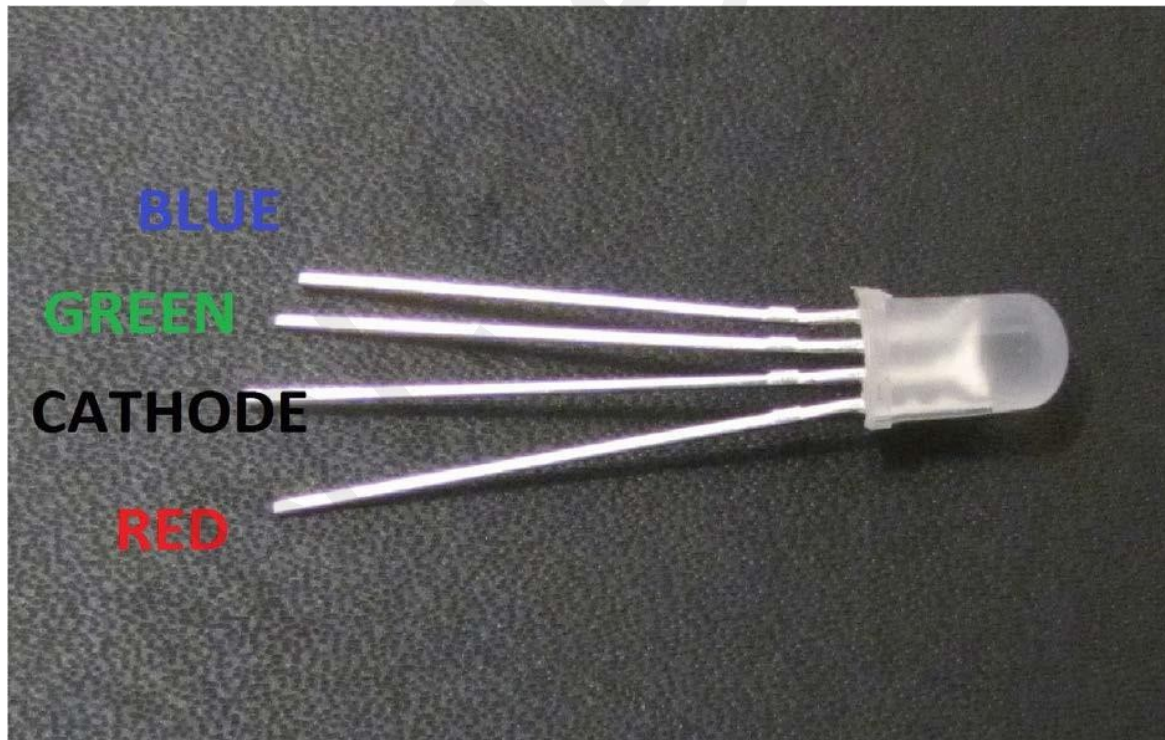
Úvod do složky

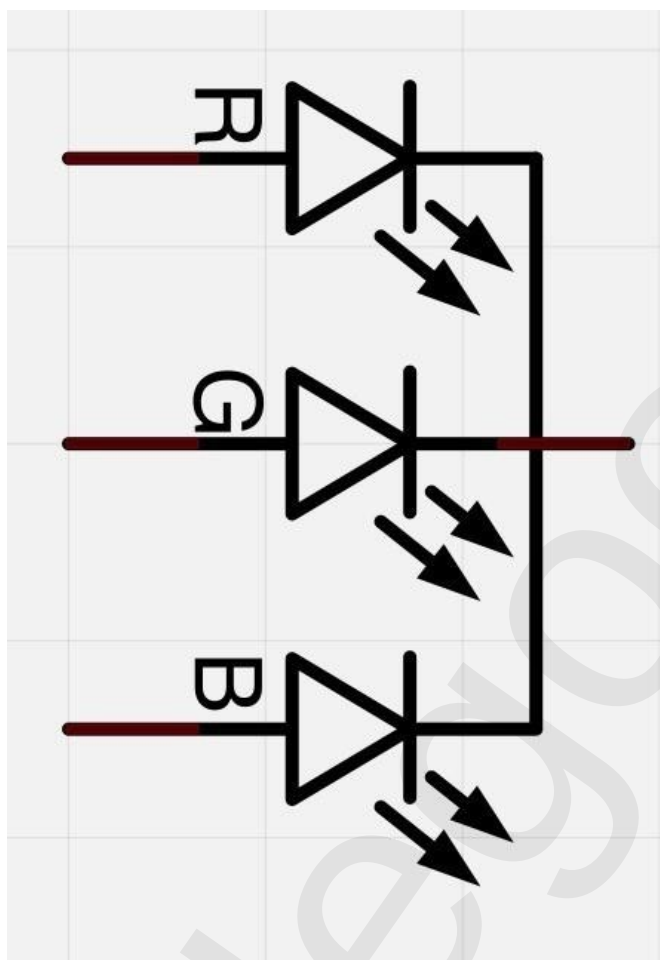
RGB:

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 UNO R3 funkci `analogWrite`, kterou můžete použít s piny označenými symbolem ~ a vyvést tak na příslušné LED diody proměnné množství energie.

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 k "-" (zem). Pokud ji připojíte opačně, LED dioda nebude svítit.

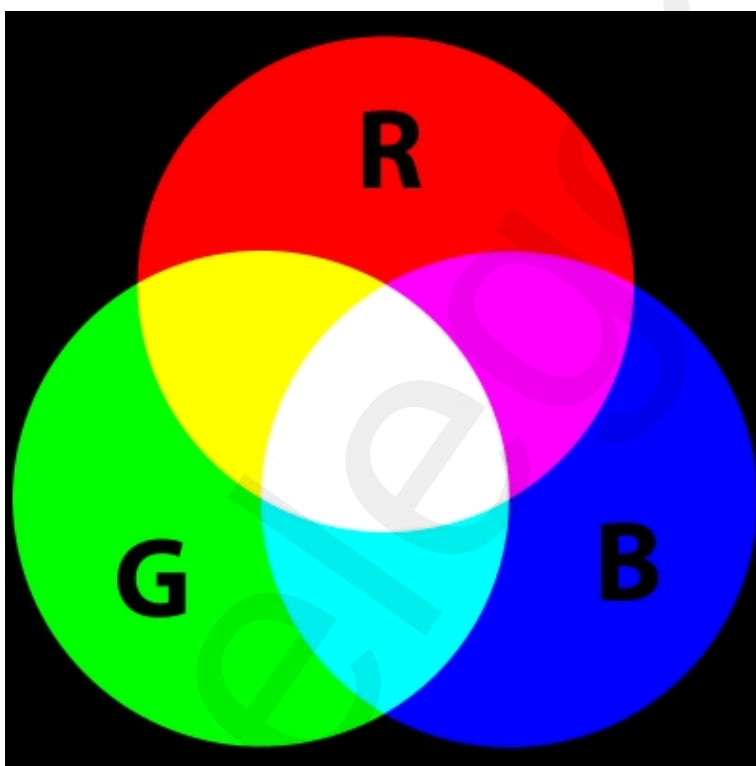
Společným záporným připojením pouzdra LED je druhý pin zploché strany. Je také nejdelší ze čtyř vodičů a bude připojen k zemi.

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

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.

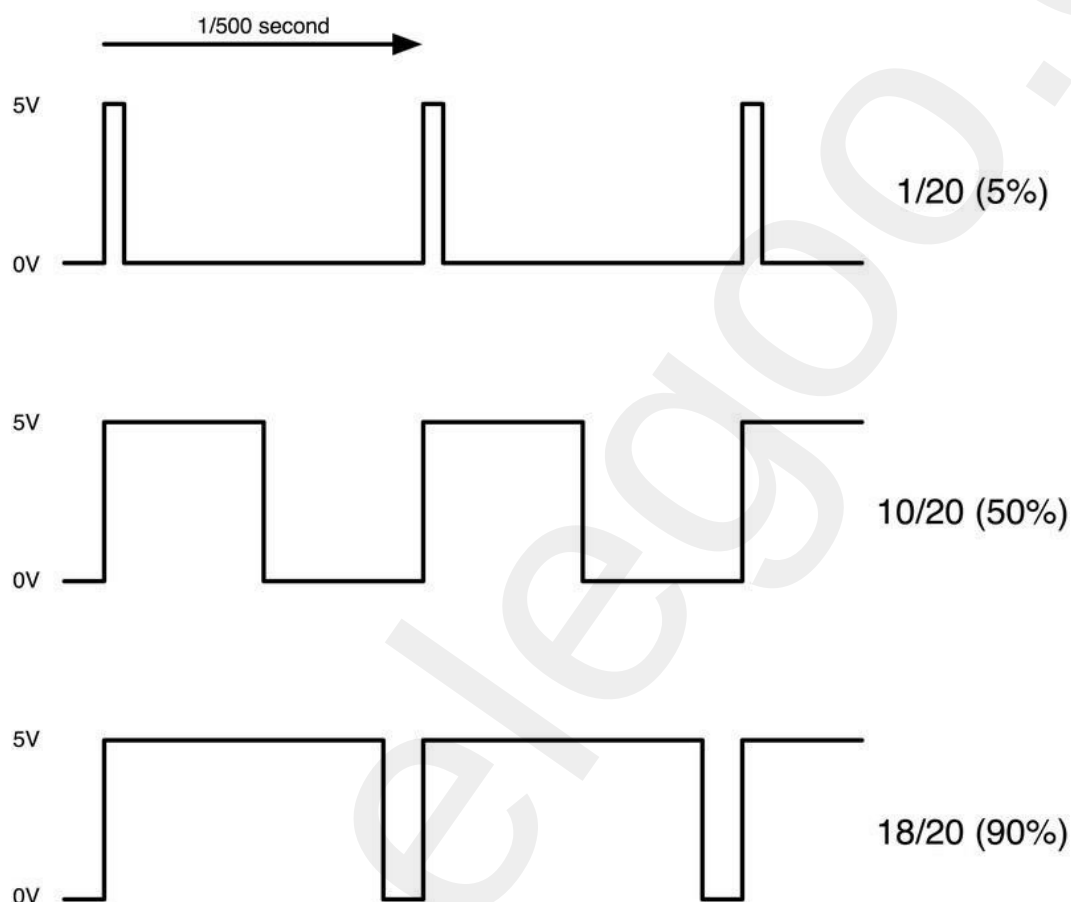
Jas každé z červených, zelených a modrých částí diody LED můžeme ovládat samostatně, takže je možné namíchat libovolnou barvu.

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 UNO.

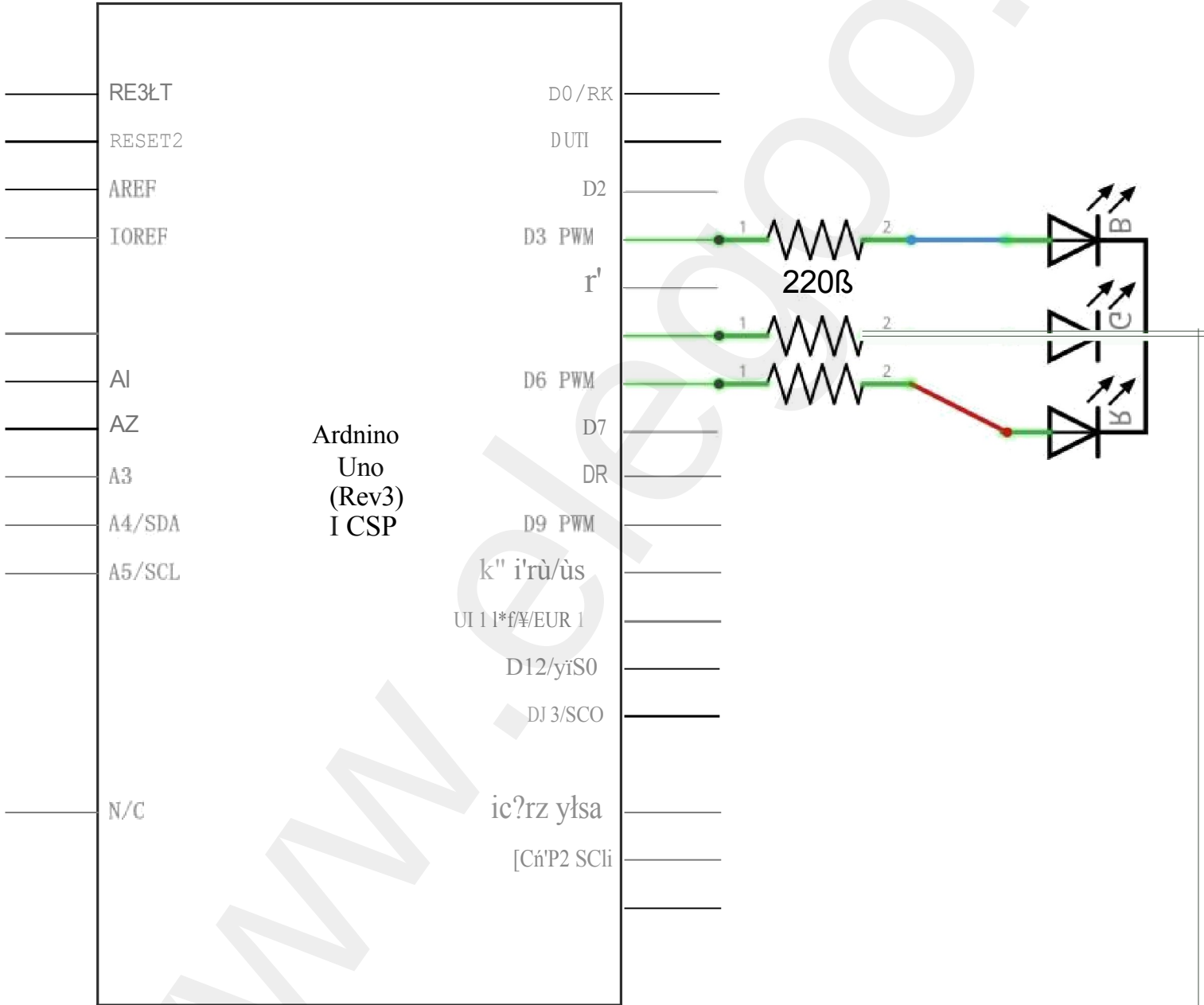


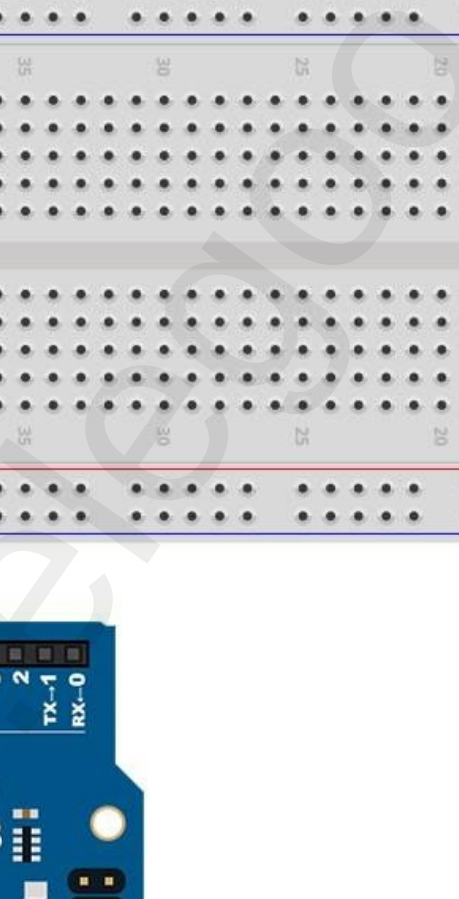
Zhruba každou 1/500 sekundy se na výstupu PWM objeví puls. Délka tohoto impulsu se řídí funkcí 'analogWrite'. Funkce 'analogWrite(0)' tedy nevytvoří vůbec žádný impuls a funkce 'analogWrite(255)' vytvoří impuls, který trvá celou dobu, dokud nepřijde další impuls, takže výstup je vlastně stále zapnutý.

Zadáme-li v analogWrite hodnotu, která je někde mezi 0 a 255, vytvoříme puls. Pokud je výstupní impuls vysoký pouze po dobu 5 % času, pak cokoli, 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.

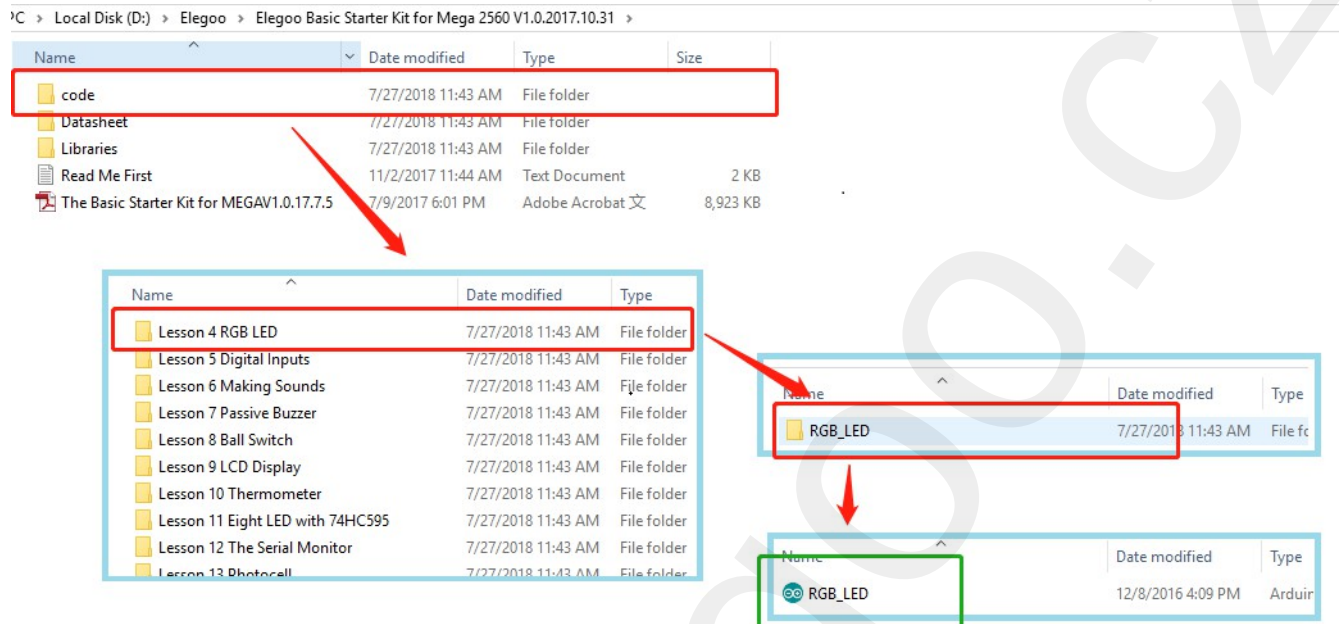
Připojení
Schéma





Kód

Po zapojení otevřete Sketch v cestě ke složce: Klepnutím na tlačítko UPLOAD nahrajte program: Výukový program > čeština > kód > Lekce 4 RGB LED > RGB_LED.



Podrobnosti o nahrávání programu v případě chyb naleznete v lekci 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 jen 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.

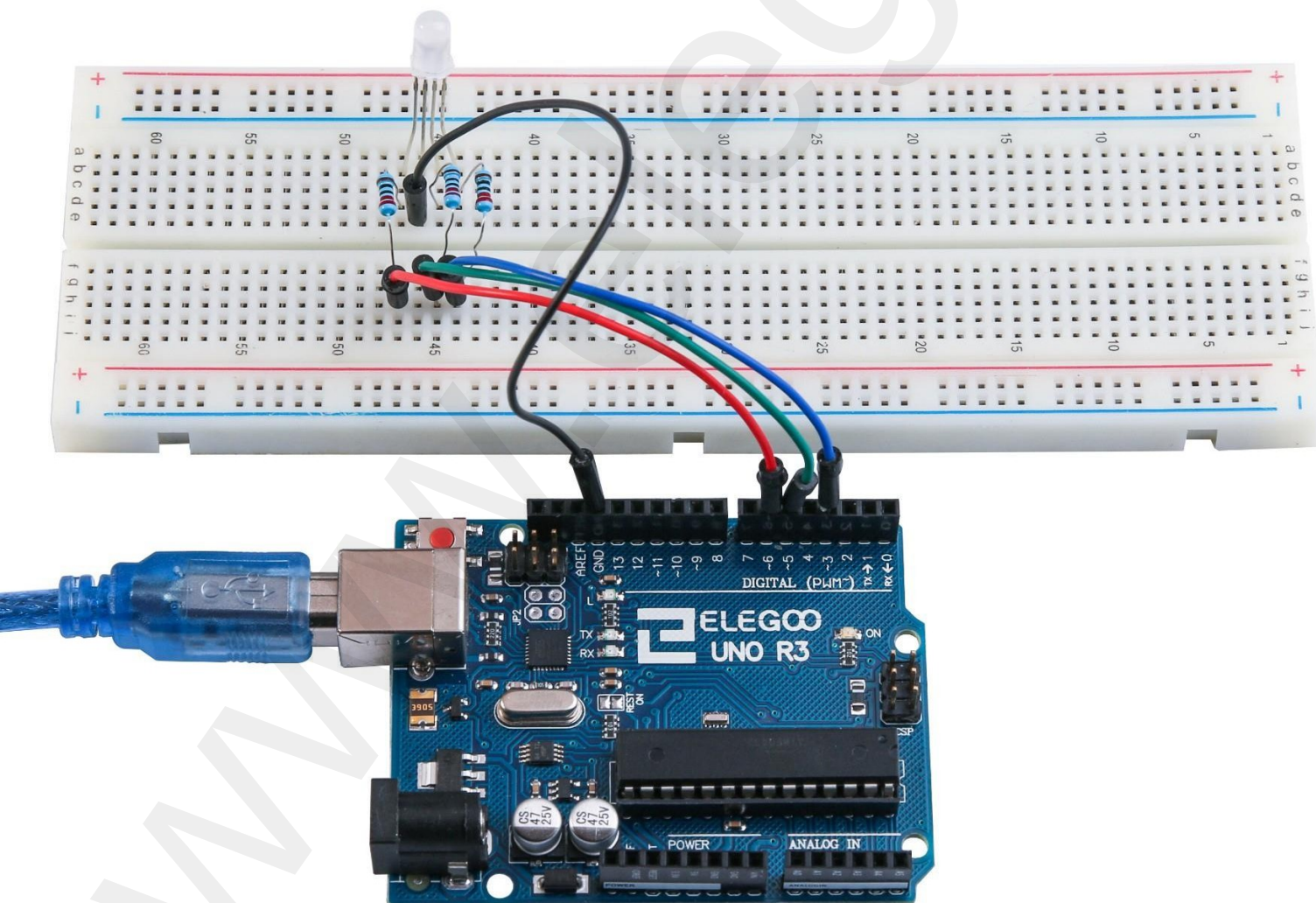
Definiční proměnné

```
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 volá 'analogWrite' pro nastavení jasu 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.

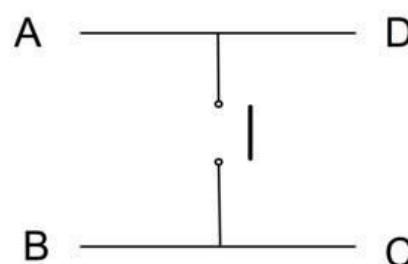
Požadovaná součást:

- (1) x Elegoo Uno R3
- (1) x 830 vazebních bodů Drátěná deska
- (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 vývody B a C propojeny dohromady, stejně jako vývody A a D.

Připojení

Schéma

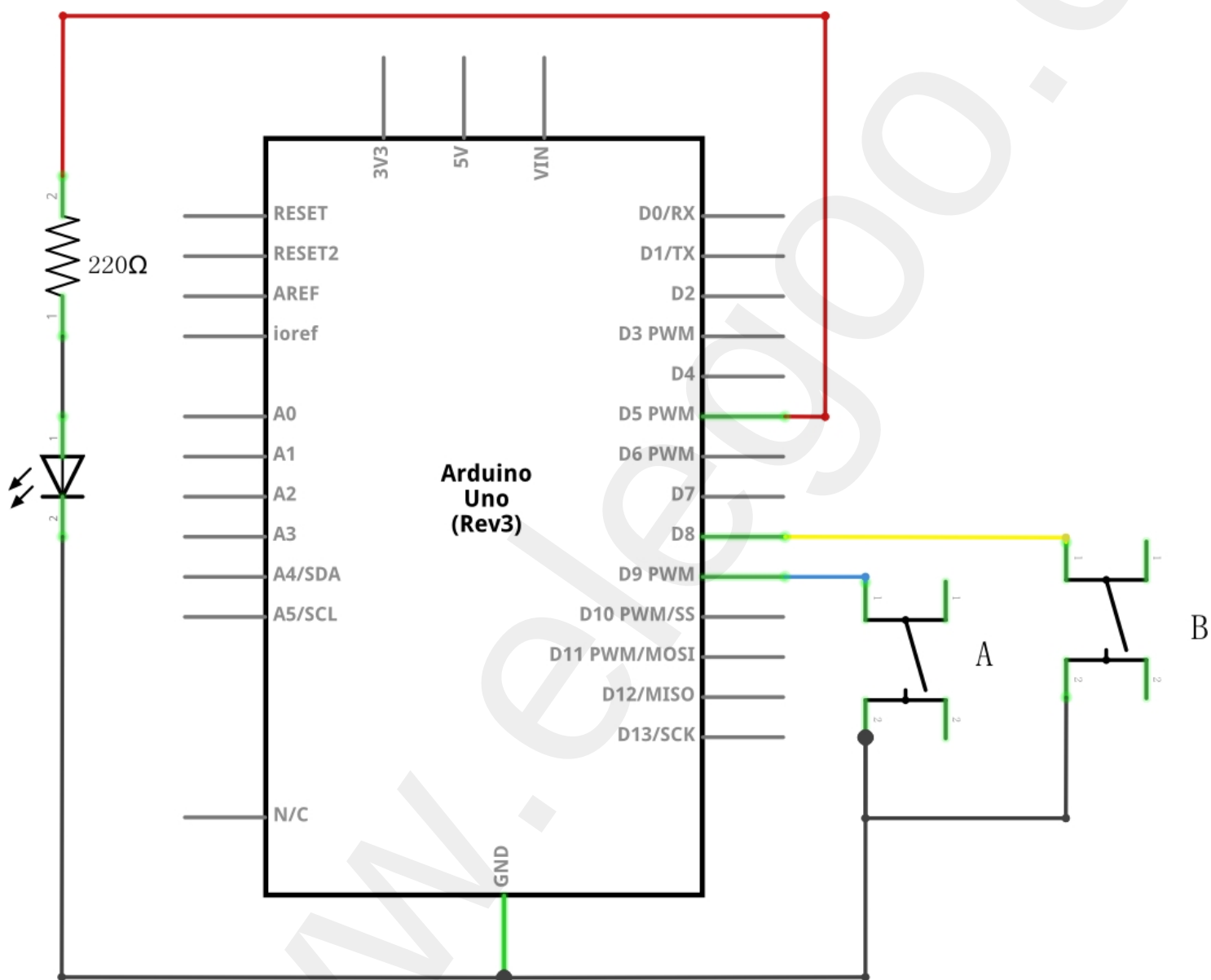
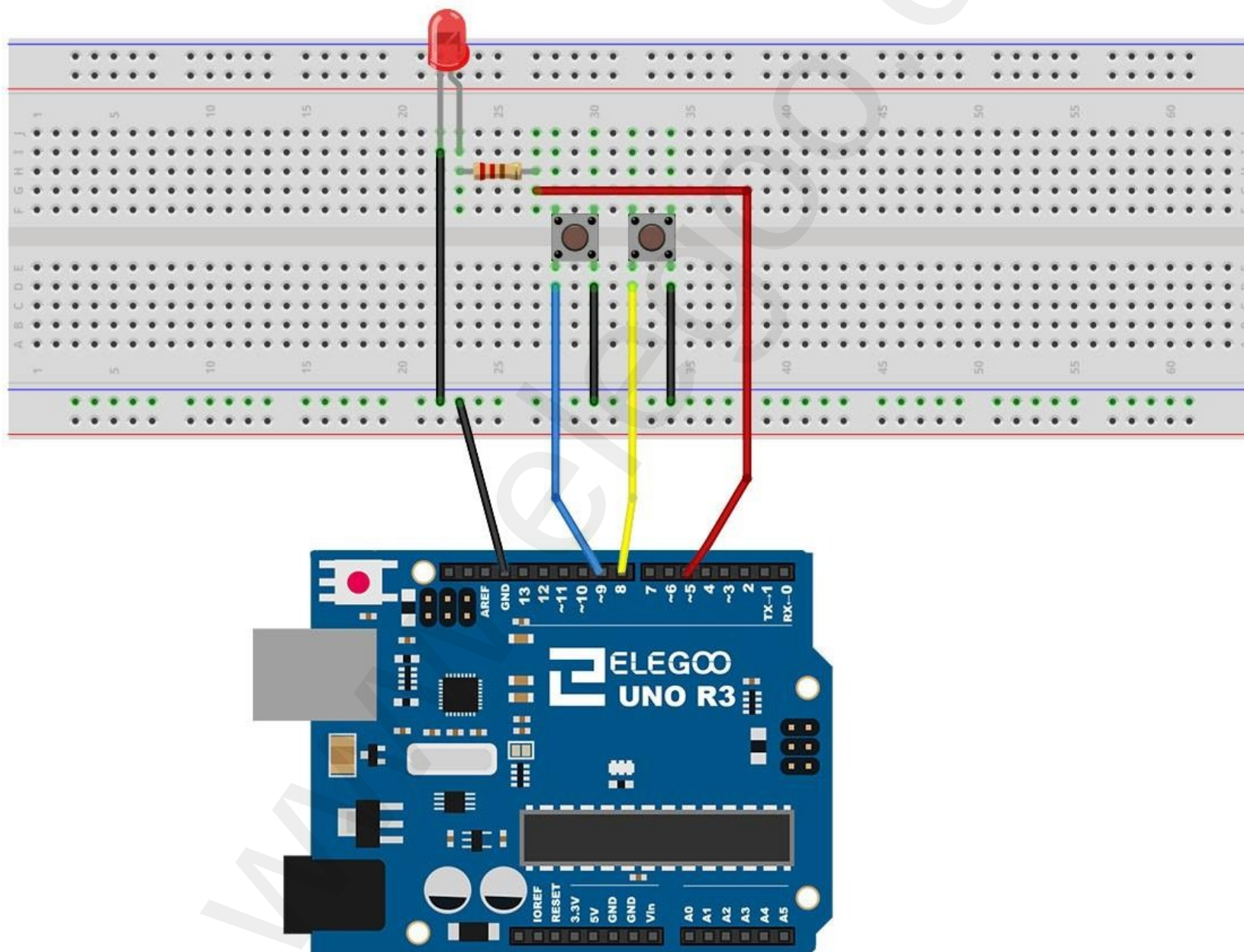


Schéma zapojení



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

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 tlačítko UPLOAD pro nahrání programu. Pokud se zobrazí chybové hlášení, podívejte se do lekce 2, kde najdete podrobnosti o výukovém programu pro nahrávání programu.

Načtete náčrt na desku UNO. 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 použijeme nastavení 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 v důsledku stisknutí tlačítka.

Proto jsou přepínače připojeny k 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);
}
}

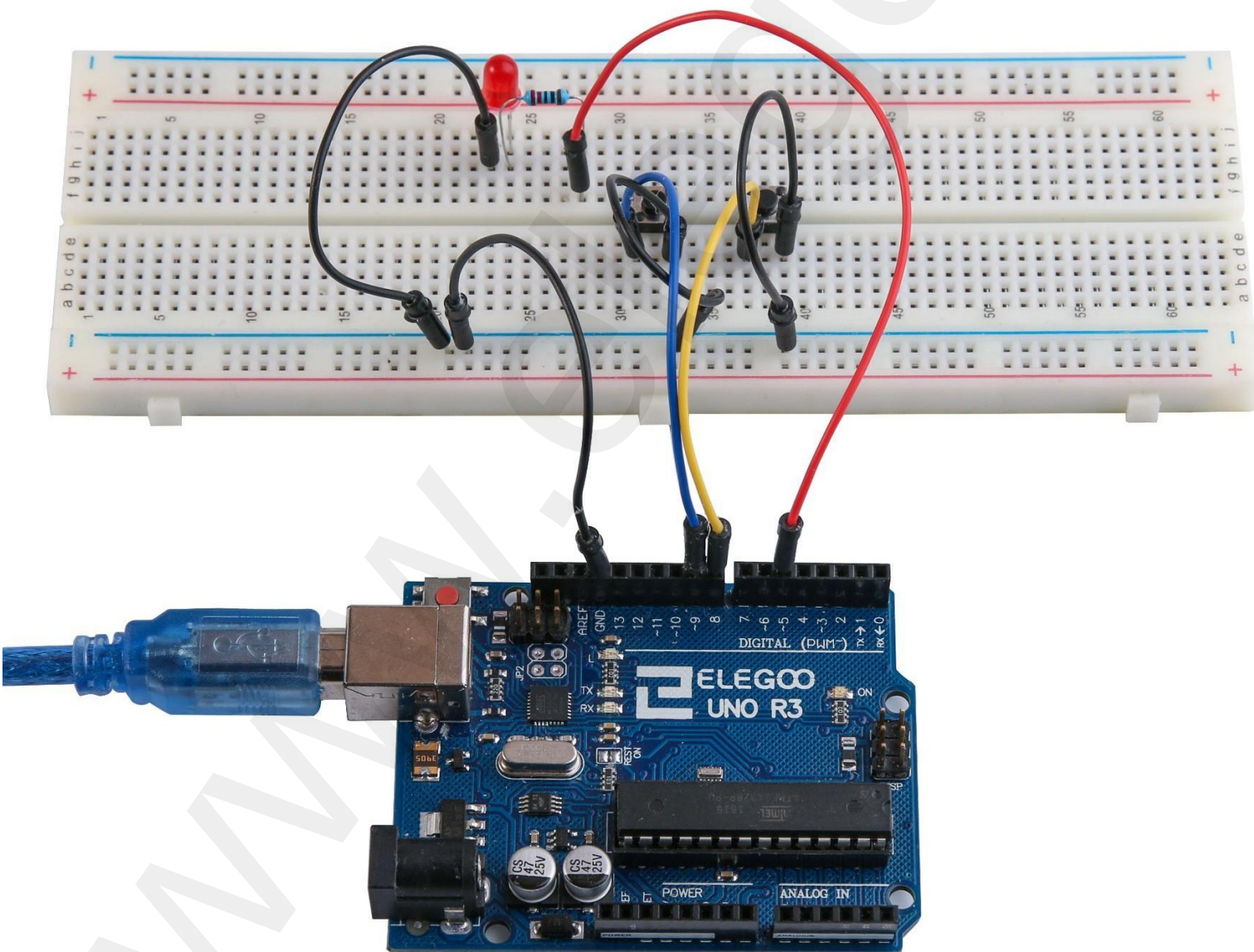
```

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 Uno 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 zdroj kmitání, takže po zapojení do elektřiny vydává zvuk. Pasivní bzučák takový zdroj nemá, takže při použití stejnosměrných signálů nebude cvrlikat; 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 zabudováno více oscilačních obvodů.



Připojení
Schéma

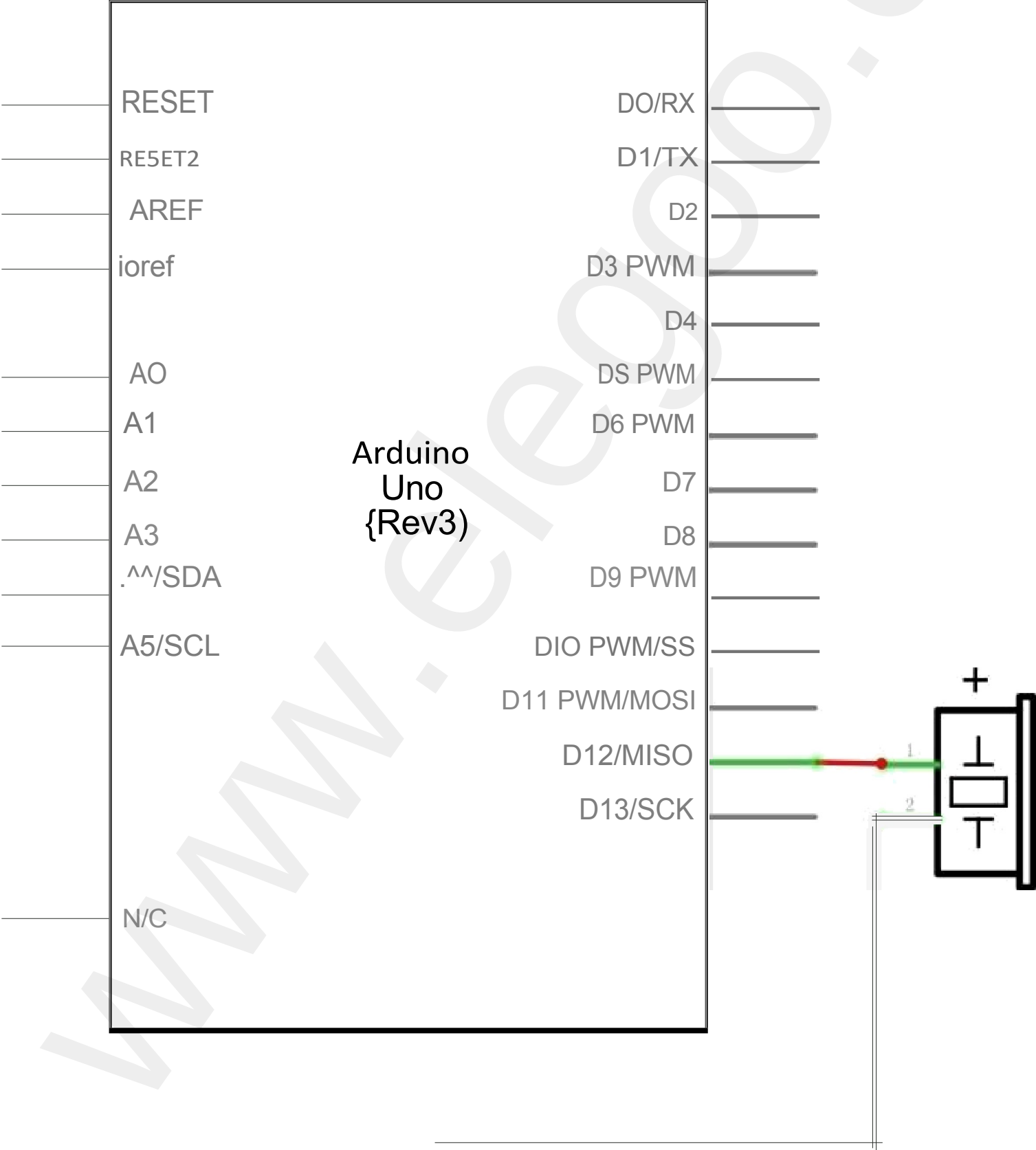
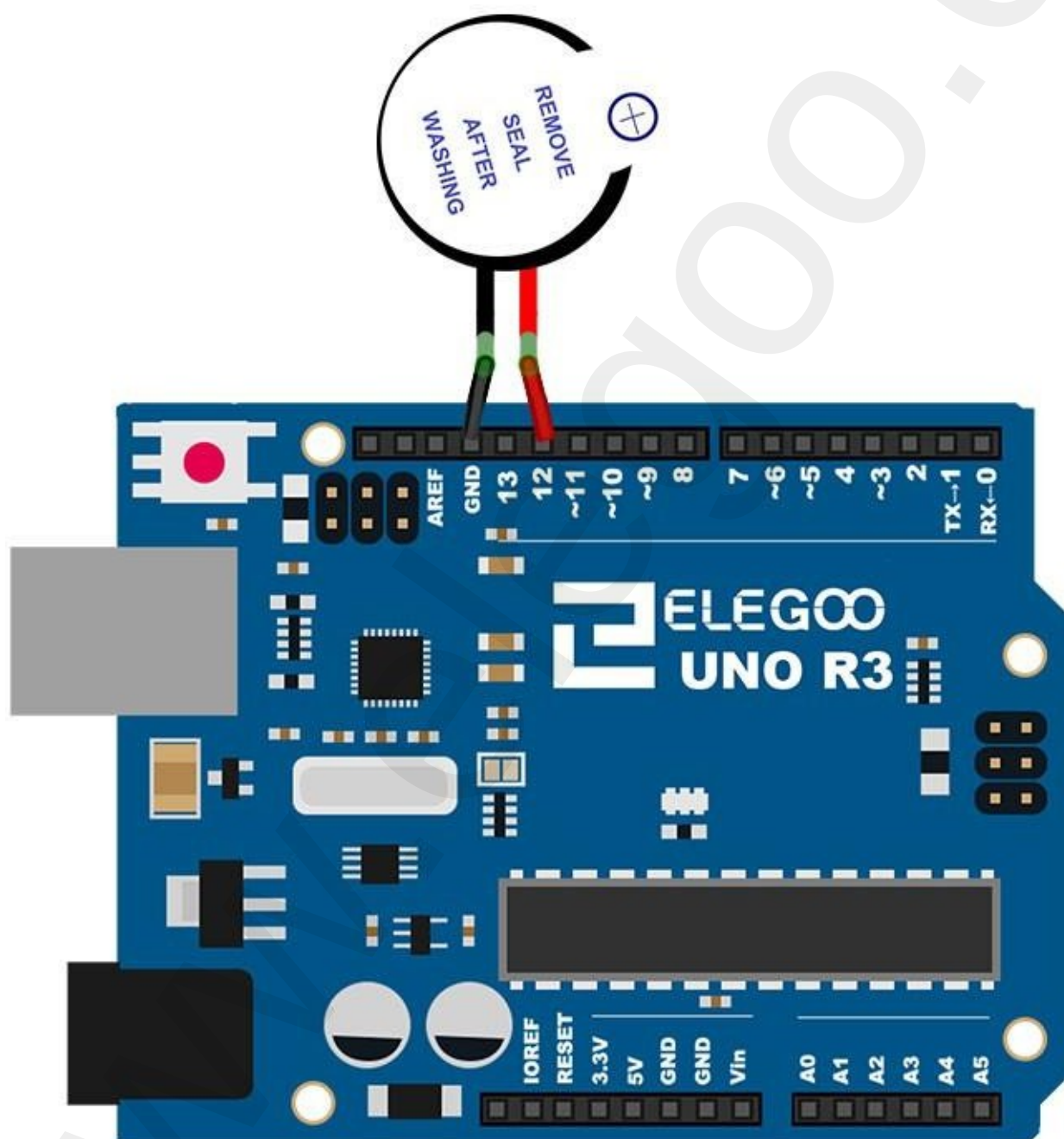


Schéma zapojení



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 nějaké chyby.

Příklad obrázku



Lekce 7 Přepínač koulí

Přehled

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

Požadovaná součást:

- (1) x Elegoo Uno 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 z pochopitelných 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. Jeden konec dutiny má dva vodivé prvky (póly). Když je snímač orientován tak, že je tento konec směrem dolů, hmota se na póly naválí, zkratuje je a 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 dalších obvodů.

Připojení
Schéma

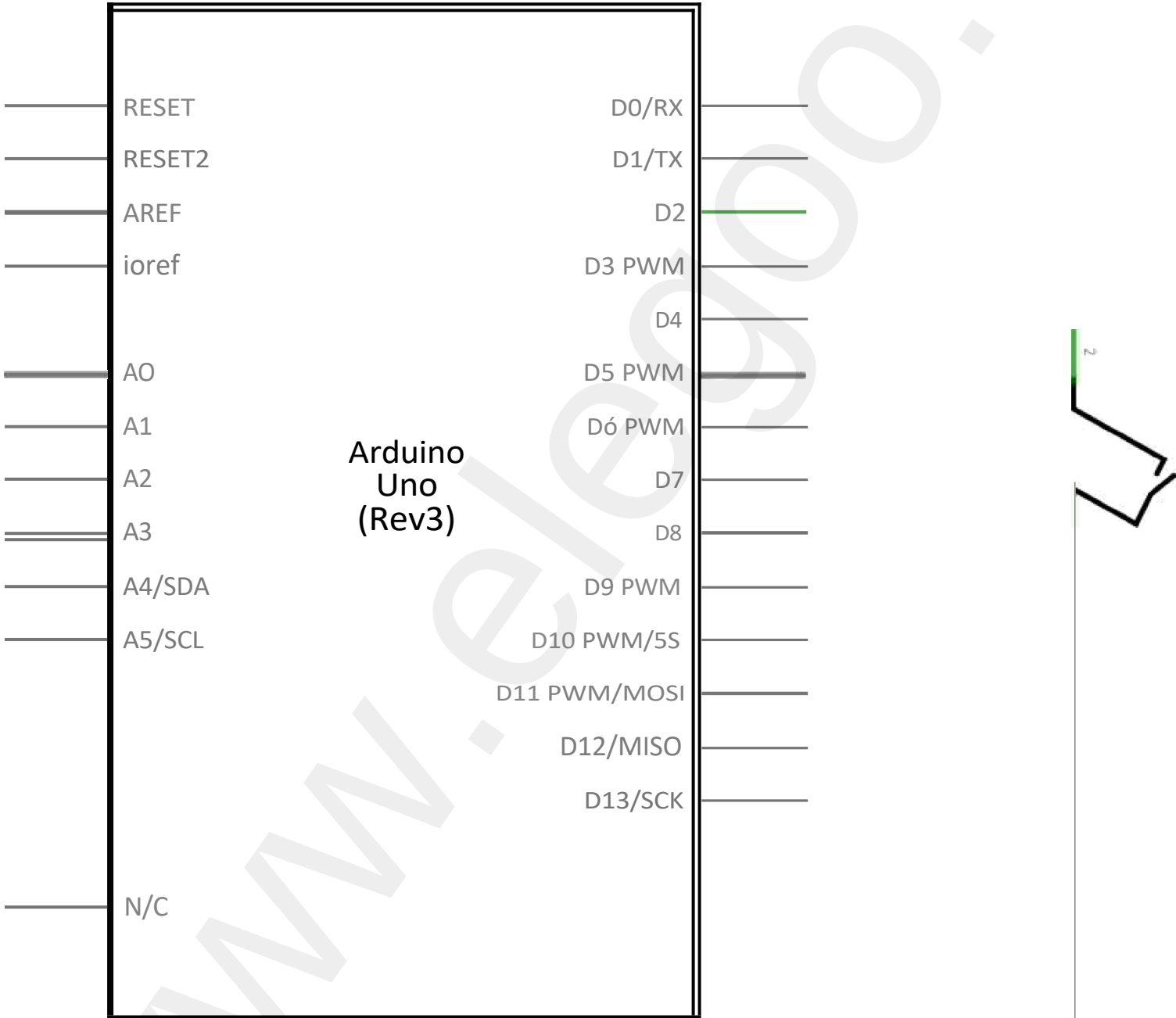
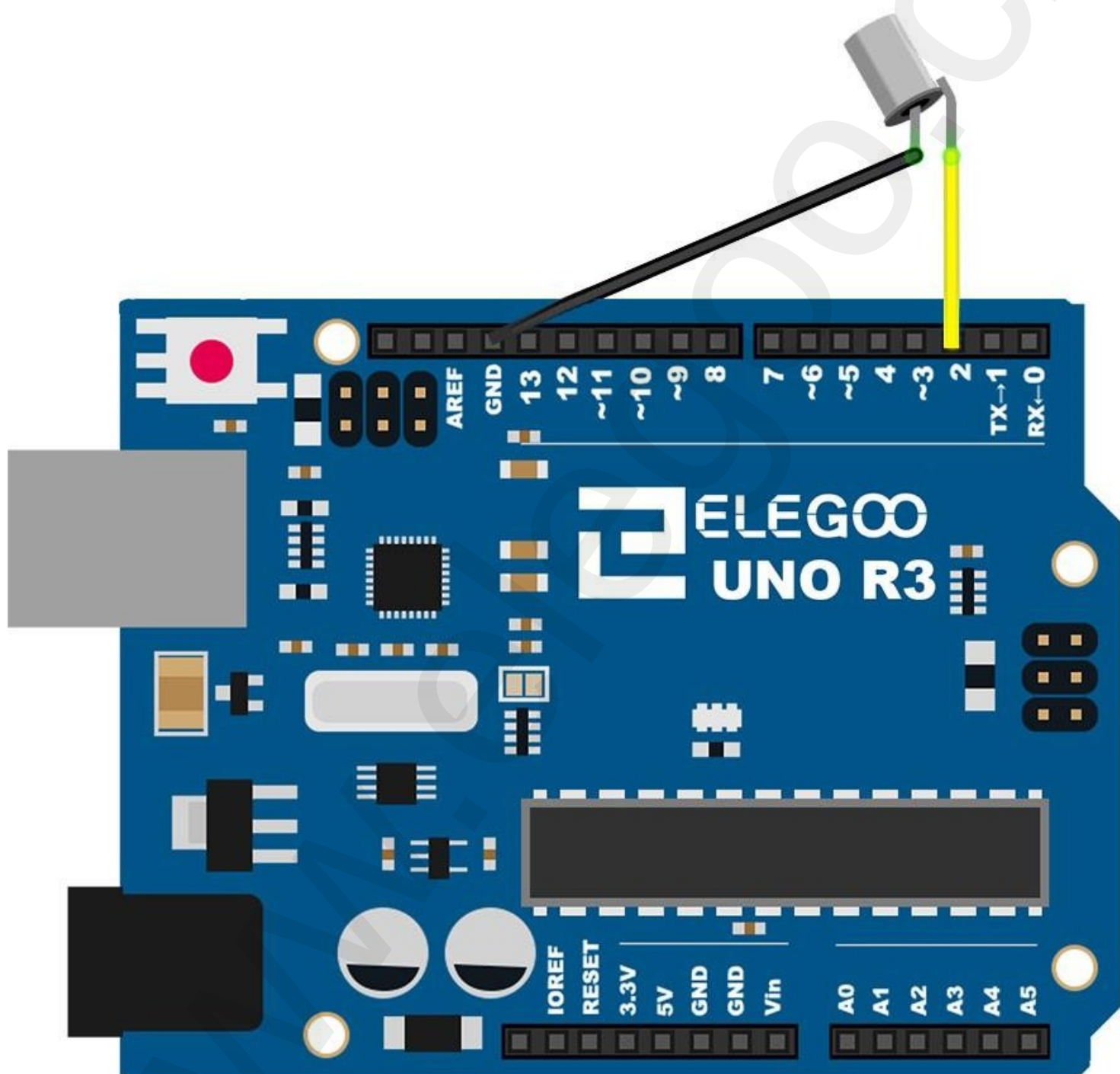


Schéma zapojení



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 8 Osm LED s 74HC595

Přehled

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

Ačkoli byste mohli zapojit osm LED diod, každou s rezistorem na jeden pin UNO, rychle by vám začaly docházet piny na UNO. Pokud nemáte k UNO 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. Takže místo toho použijete čip s názvem 74HC595 Serial to Parallel Converter. Tento čip má osm výstupů (perfektní) a tři vstupy, kterými do něj přivádíte data po kouskách.

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:

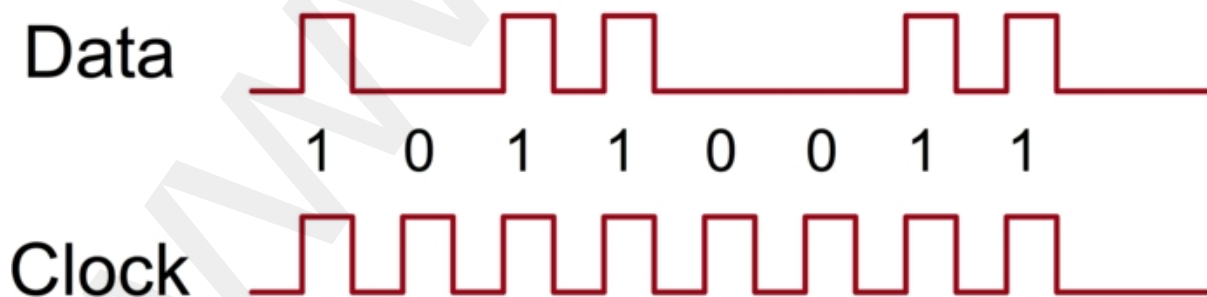
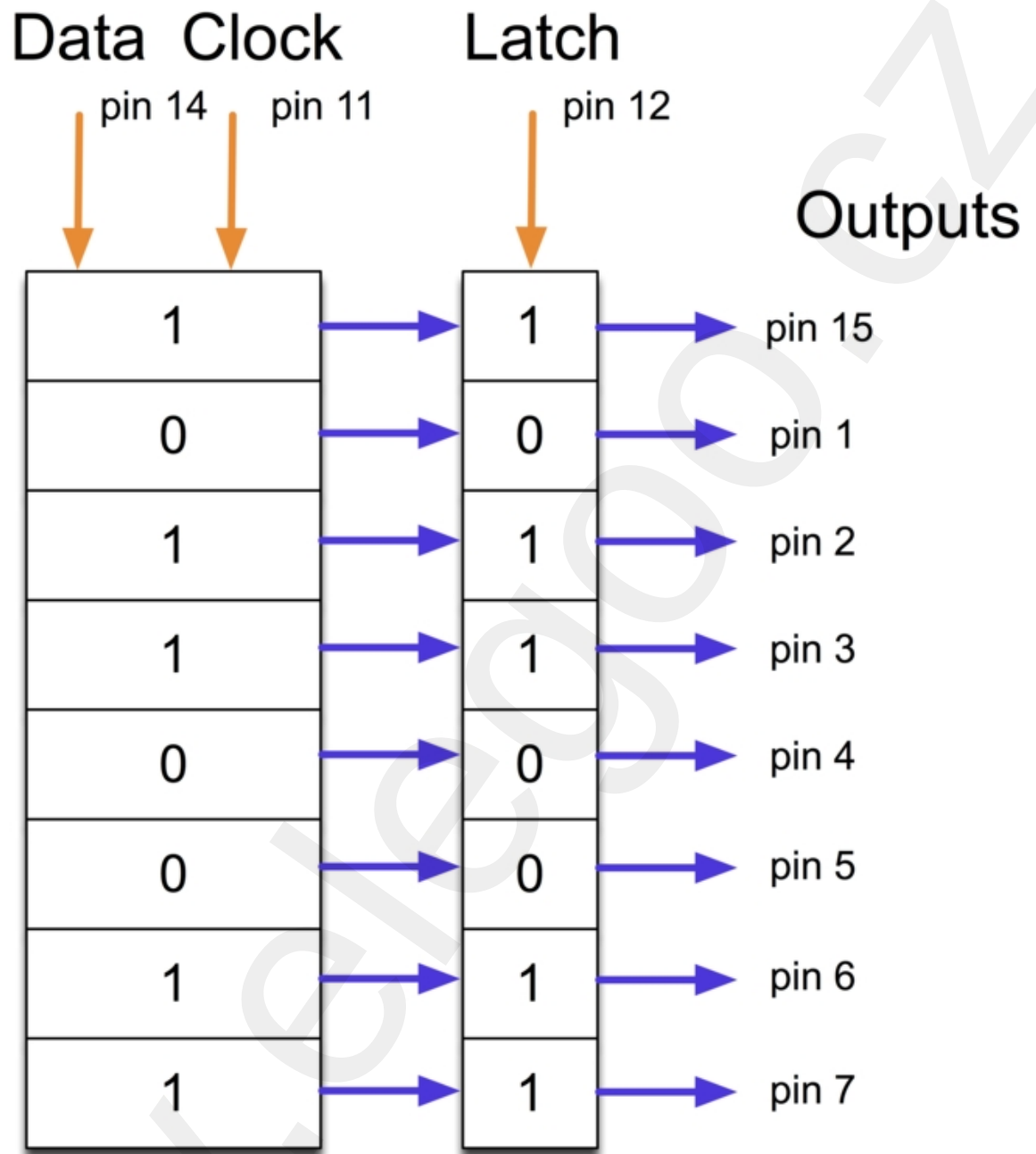
- (1) x Elegoo Uno R3
- (1) x 830 vazebních bodů chlebová deska
- (8) x LED diody
- (8) x 220 ohmů rezistorů
- (1) x 74hc595 IC
- (14) x vodiče M-M (propojovací vodiče typu Male to Male)



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.



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

Schéma

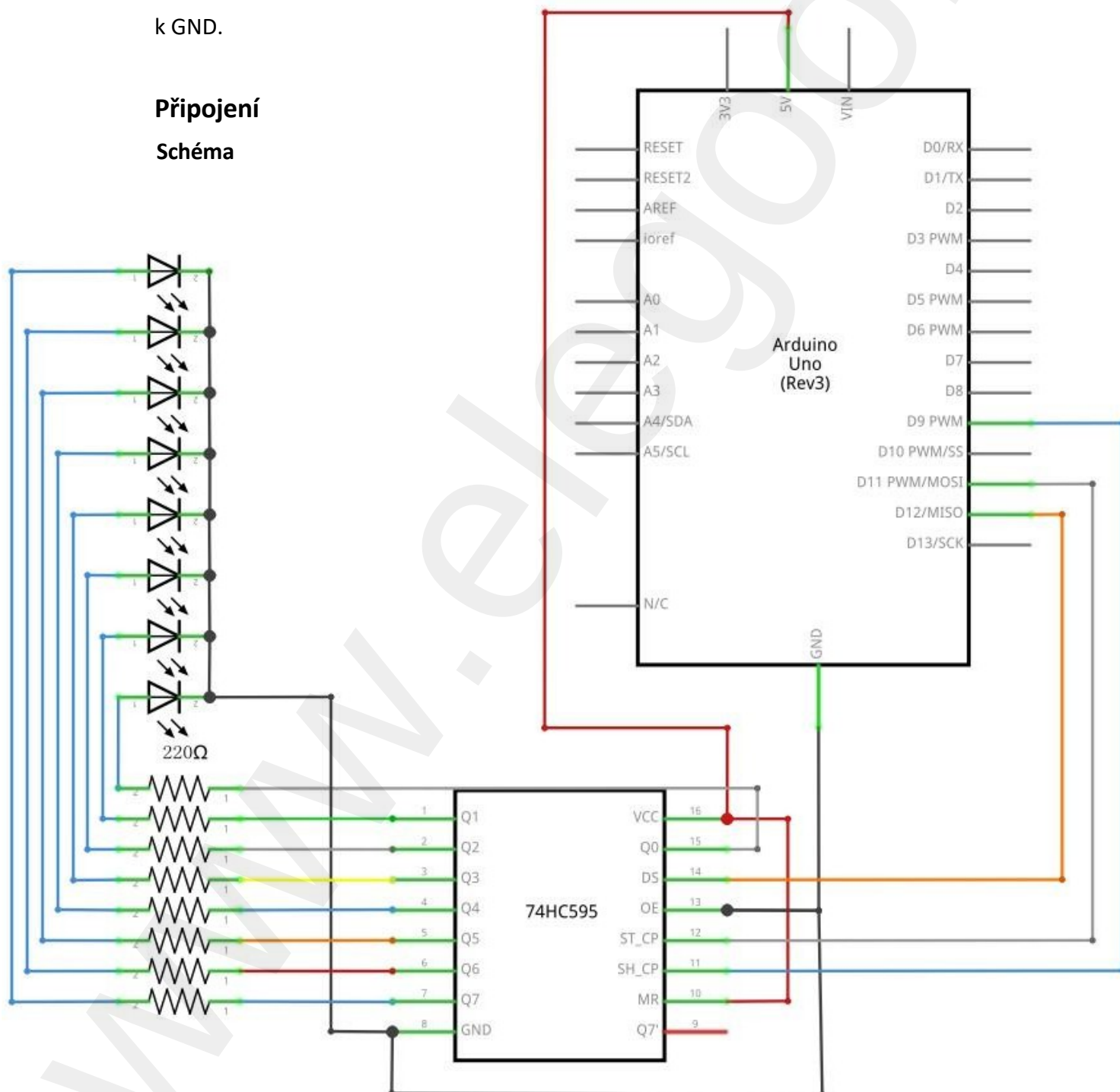
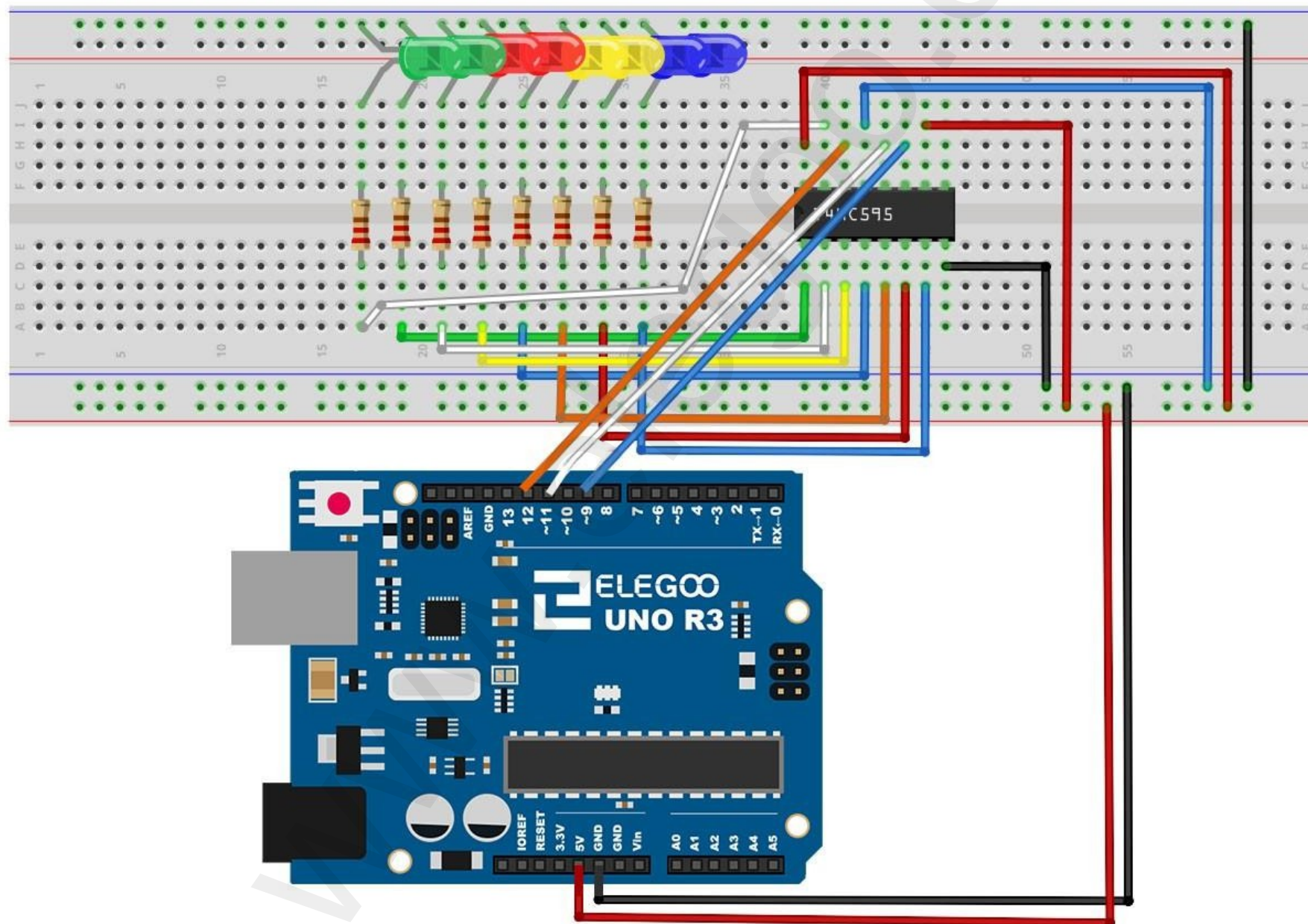


Schéma zapojení



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 UNO jde na pin #14 posuvného registru Digitál

11 z UNO jde na pin #12 posuvného registru Digitál 9 z UNO

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 UNO 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 24 Osm LED s

74HC595 a klikněte na tlačítko UPLOAD pro nahrání programu. Podrobnosti o nahrávání programu v případě chyb naleznete v lekci 2.

Nejprve definujeme tři piny, které budeme používat. Jedná se o digitální výstupy UNO, 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.

které z našich osmi LED diod svítí

nebo nesvítí. `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. Fungováním 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 půlvtéřinové 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 UNO 'shiftOut' a teprve poté nastaví latchPin opět 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 chcete začít. Začneme pravým bitem, který se označuje jako "nejméně významný bit".

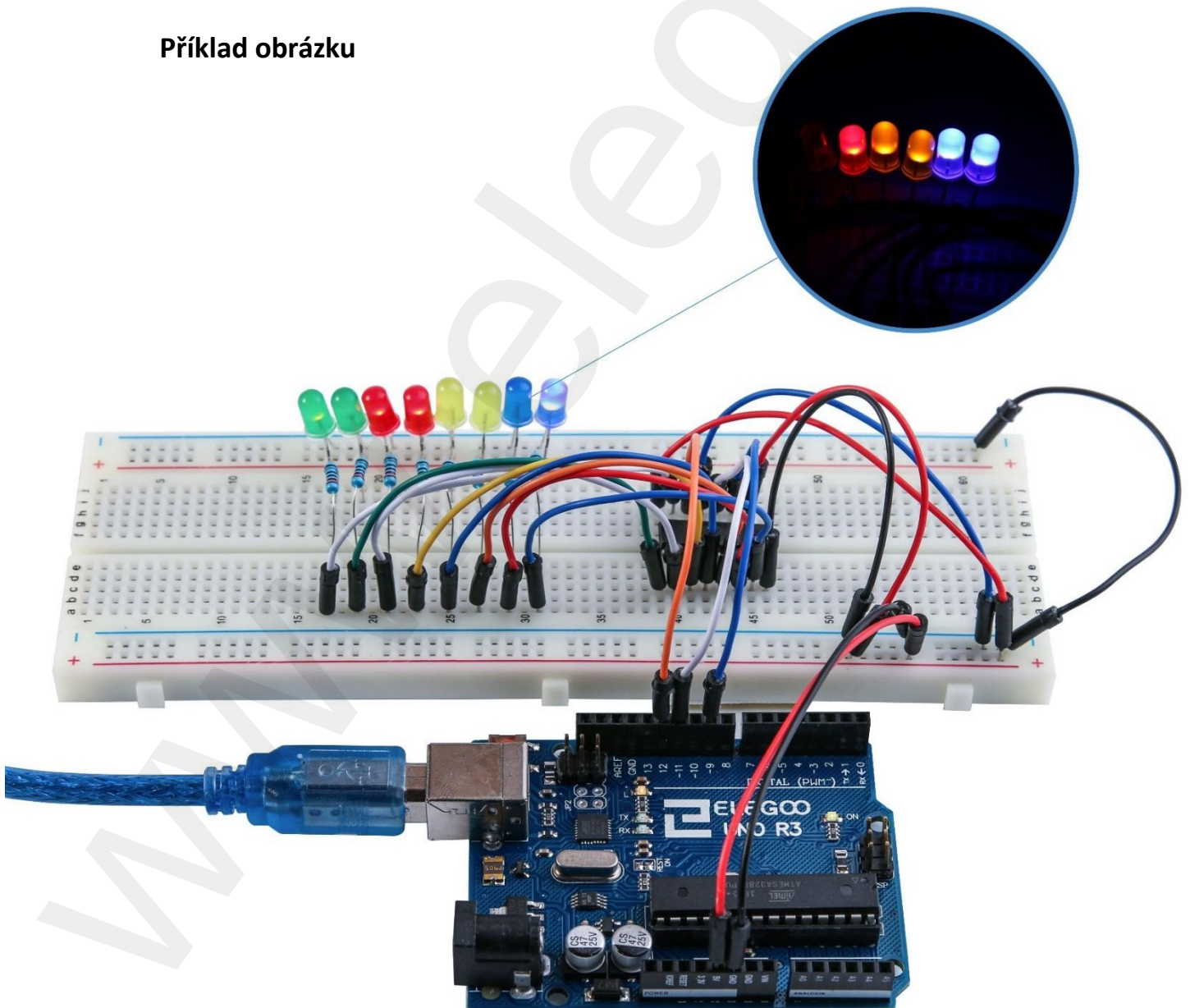
Bit' (LSB).

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 aktualizovaly skutečné LED diody.

Příklad obrázku



Lekce 9 Sériový monitor

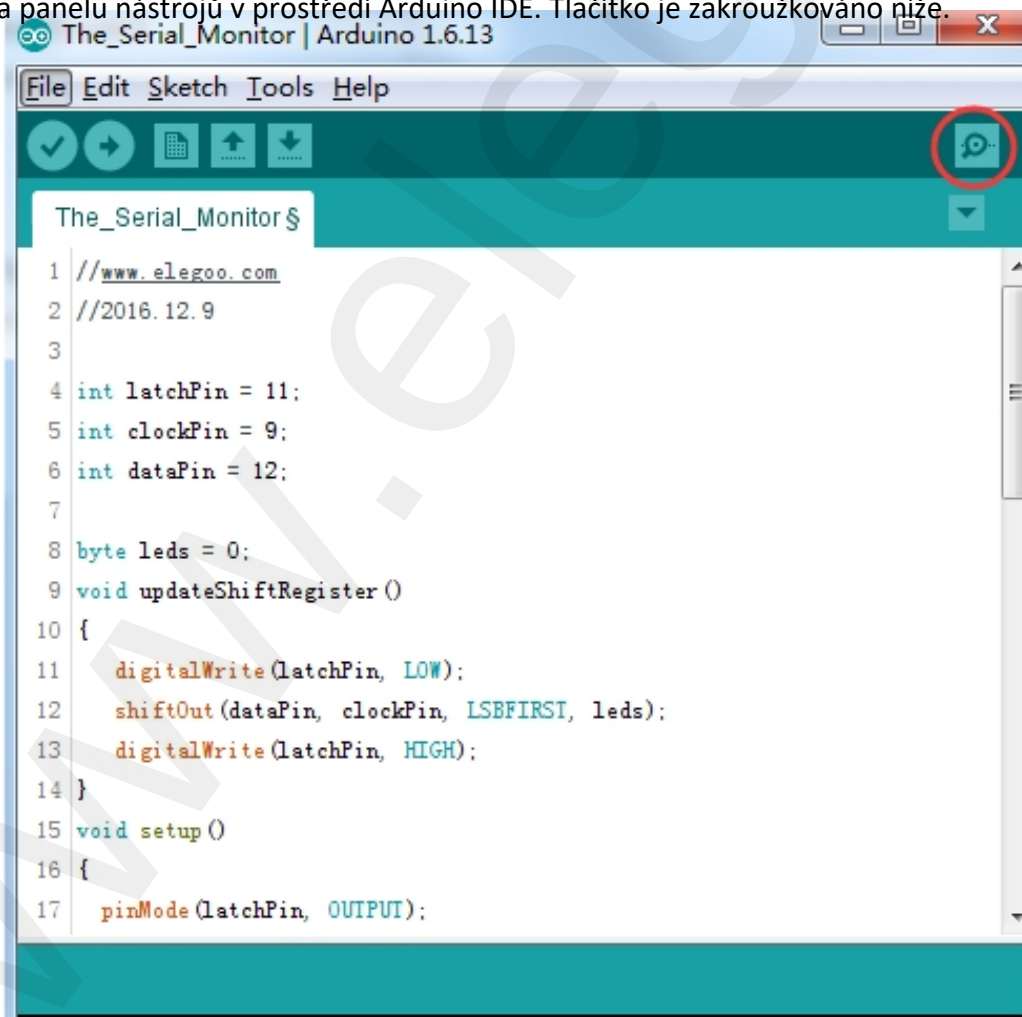
Přehled

V této lekci navážete na lekci 8 a přidáte možnost ovládat LED diody z počítače pomocí sériového monitoru Arduino. Sériový monitor je "pojítkem" mezi počítačem a vaším UNO. Umožňuje odesílat a přijímat textové zprávy, což se hodí při ladění a také při ovládání UNO z klávesnice! Například budete moci z počítače posílat příkazy k zapnutí LED diod. V této lekci použijete úplně stejné součástky a podobné uspořádání desky jako v lekci 8. Pokud jste tak tedy ještě neučinili, postupujte nyní podle lekce 8.

Přijaté kroky

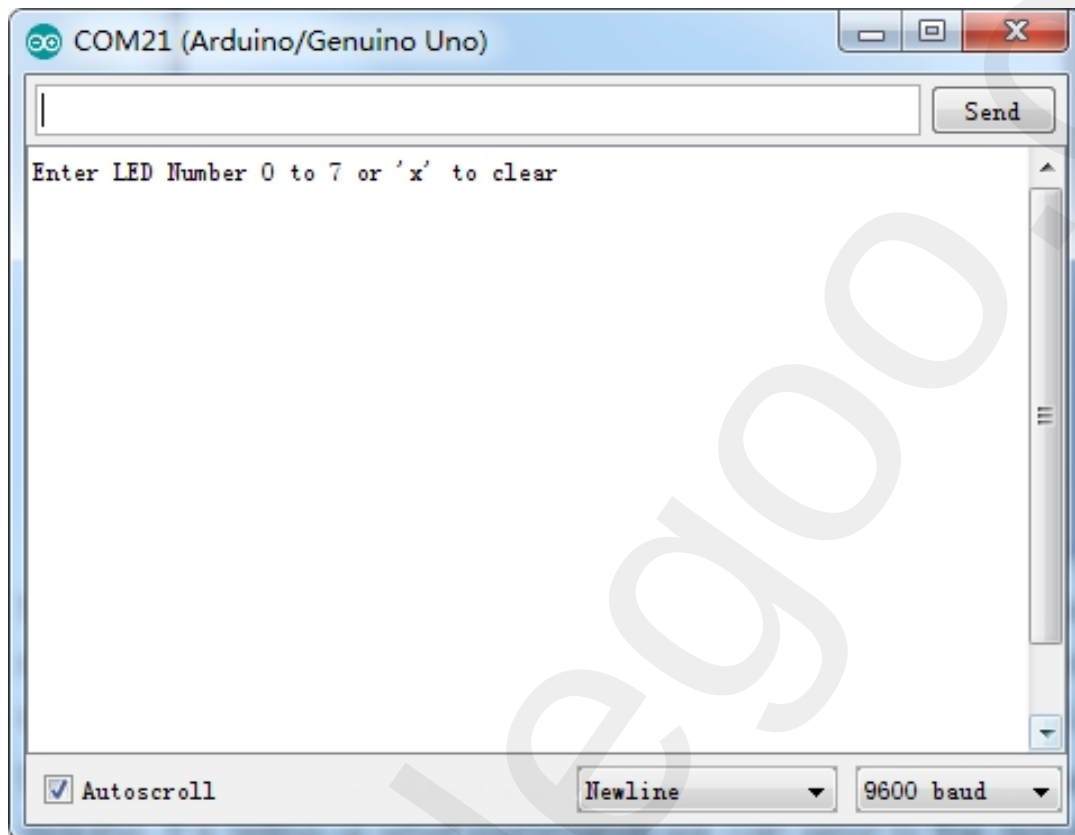
Po nahrání tohoto náčrtu do zařízení UNO klikněte na pravý krajní roh.

na panelu nástrojů v prostředí Arduino IDE. Tlačítko je zakroužkovááno níže.



Otevře se následující okno.

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

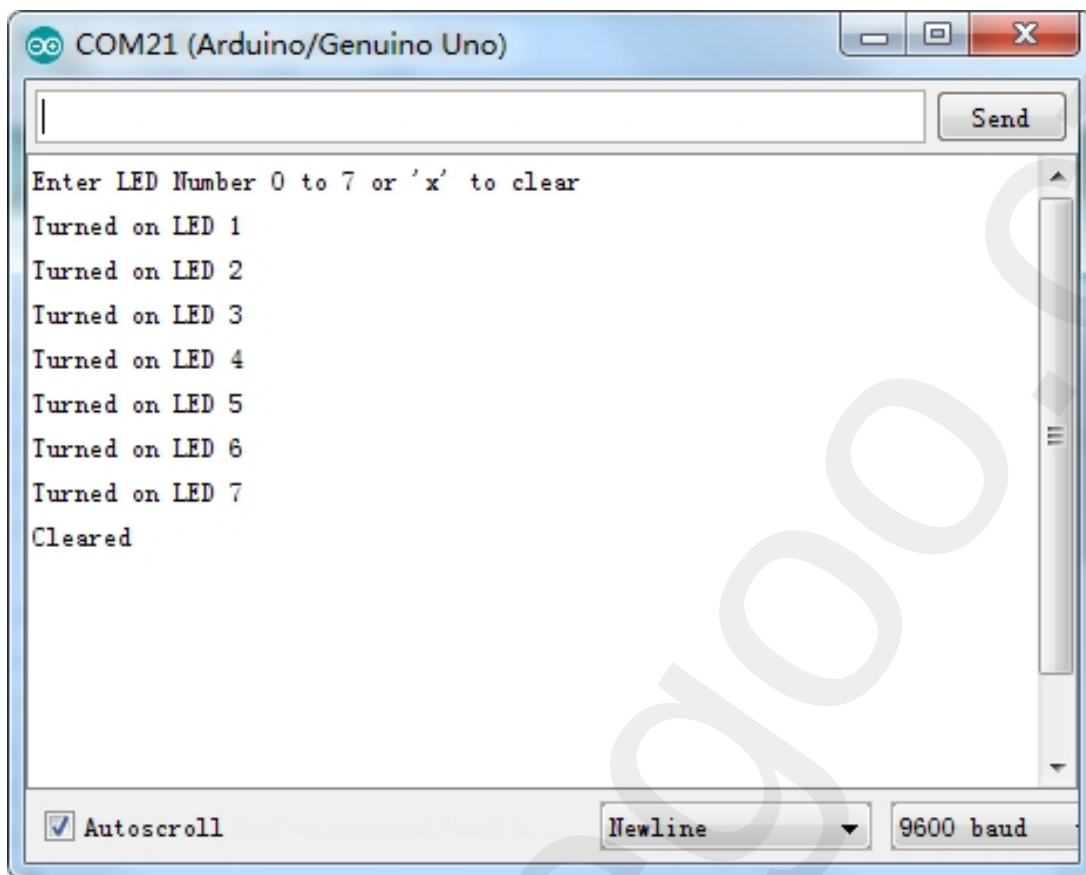


Toto okno se nazývá Serial Monitor a je součástí softwaru Arduino IDE. Jeho úkolem je umožnit odesílání zpráv z počítače na desku UNO (přes USB) a také přijímání zpráv z UNO.

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 UNO se zobrazí potvrzovací zpráva. Sériový monitor se zobrazí podle následujícího obrázku.



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

Kód

Po zapojení otevřete program ve složce s kódem - Lekce 25 Sériový monitor a kliknutím na **UPLOAD** program nahrajte. Podrobnosti o nahrávání programu naleznete v Lekci 2, pokud se vyskytnou nějaké chyby.

Jak se dalo očekávat, náčrtes vychází z náčrtu 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 UNO může odesílat příkazy prostřednictvím připojení USB. Hodnota 9600 se nazývá "přenosová rychlost" připojení. To je 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 UNO automaticky resetuje desku Arduino, když otevřete Serial Monitor, zatímco u Leonarda se tak neděje.

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

Veškerá akce se odehrává ve funkci 'loop': **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 UNO a jsou připravena ke zpracování. Příchozí zprávy jsou uchovávány v tzv. bufferu a Serial.available() vrací true, pokud tento buffer není prázdný.

Pokud byla zpráva přijata, přejdeme 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 jednočísli, 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 k dalšímu řádku:

```
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 při odečítání se ve skutečnosti používají hodnoty ASCII.

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 kontrolky LED a odešle potvrzovací zprávu.

Lekce 10 Fotobuňka

Přehled

V této lekci se naučíte měřit intenzitu světla pomocí analogového vstupu. Navážete na lekci 8 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 Uno R3
- (1) x 830 vazebních bodů chlebová deska
- (8) x LED diody
- (8) x 220 ohmů 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)



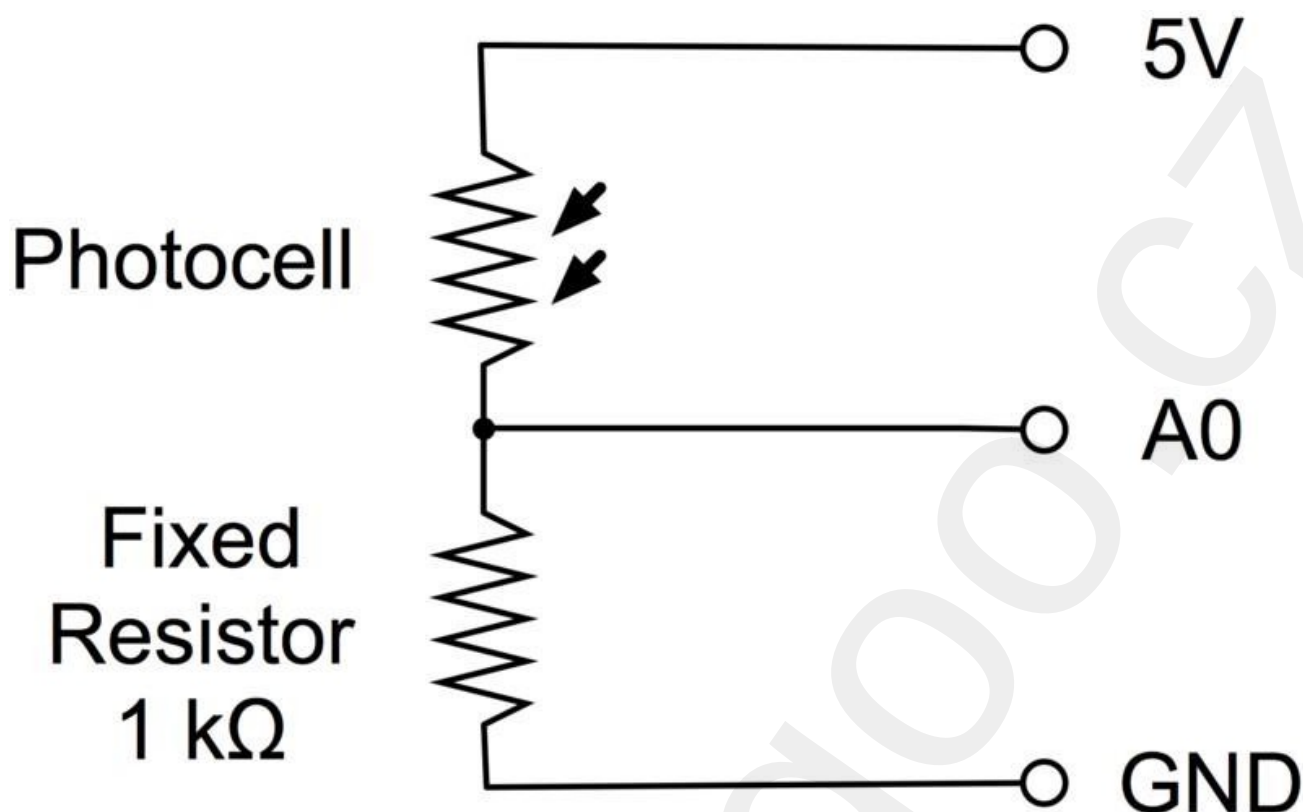
Úvod do složky

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 UNO 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.

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

Připojení Schéma

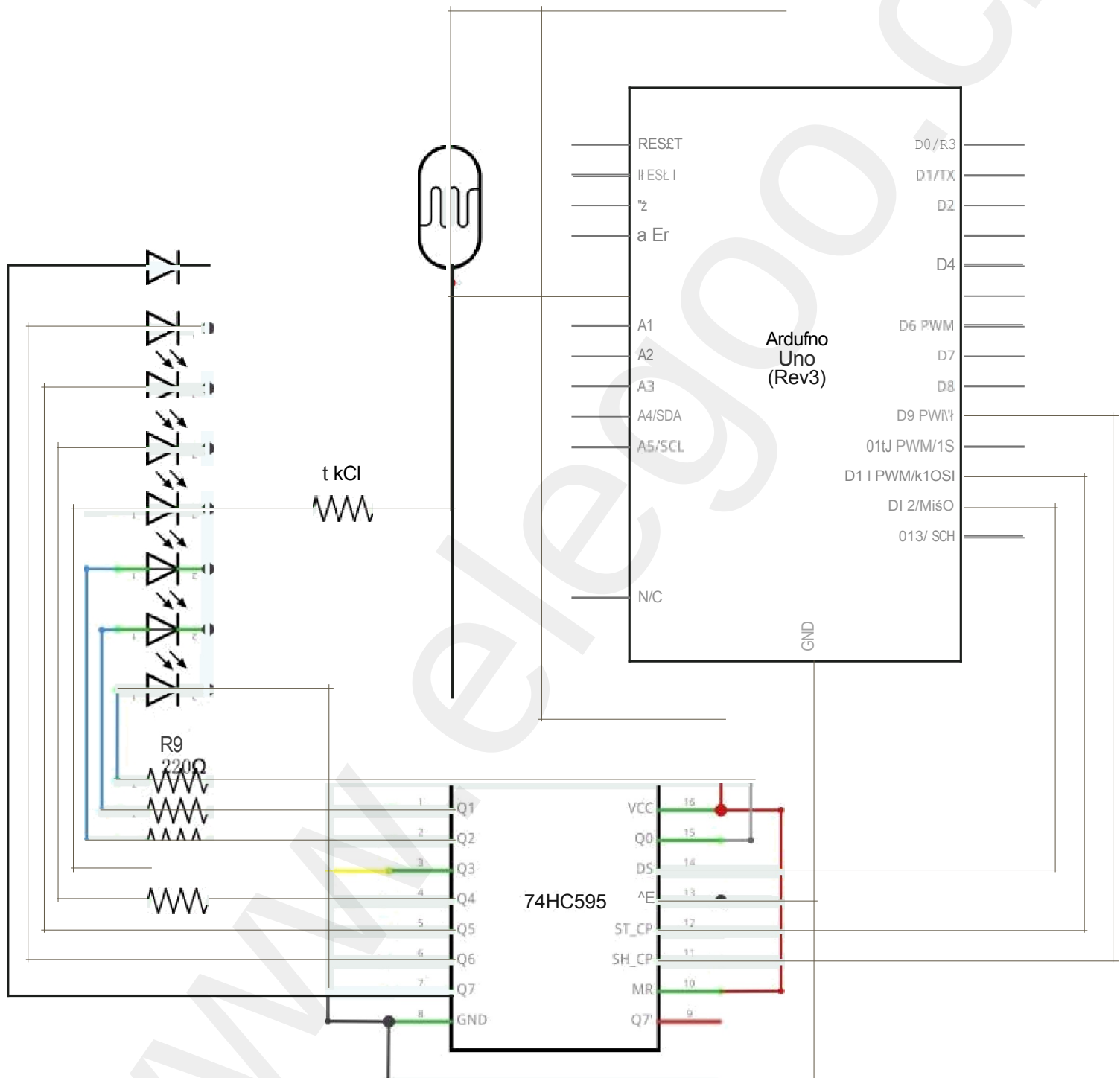
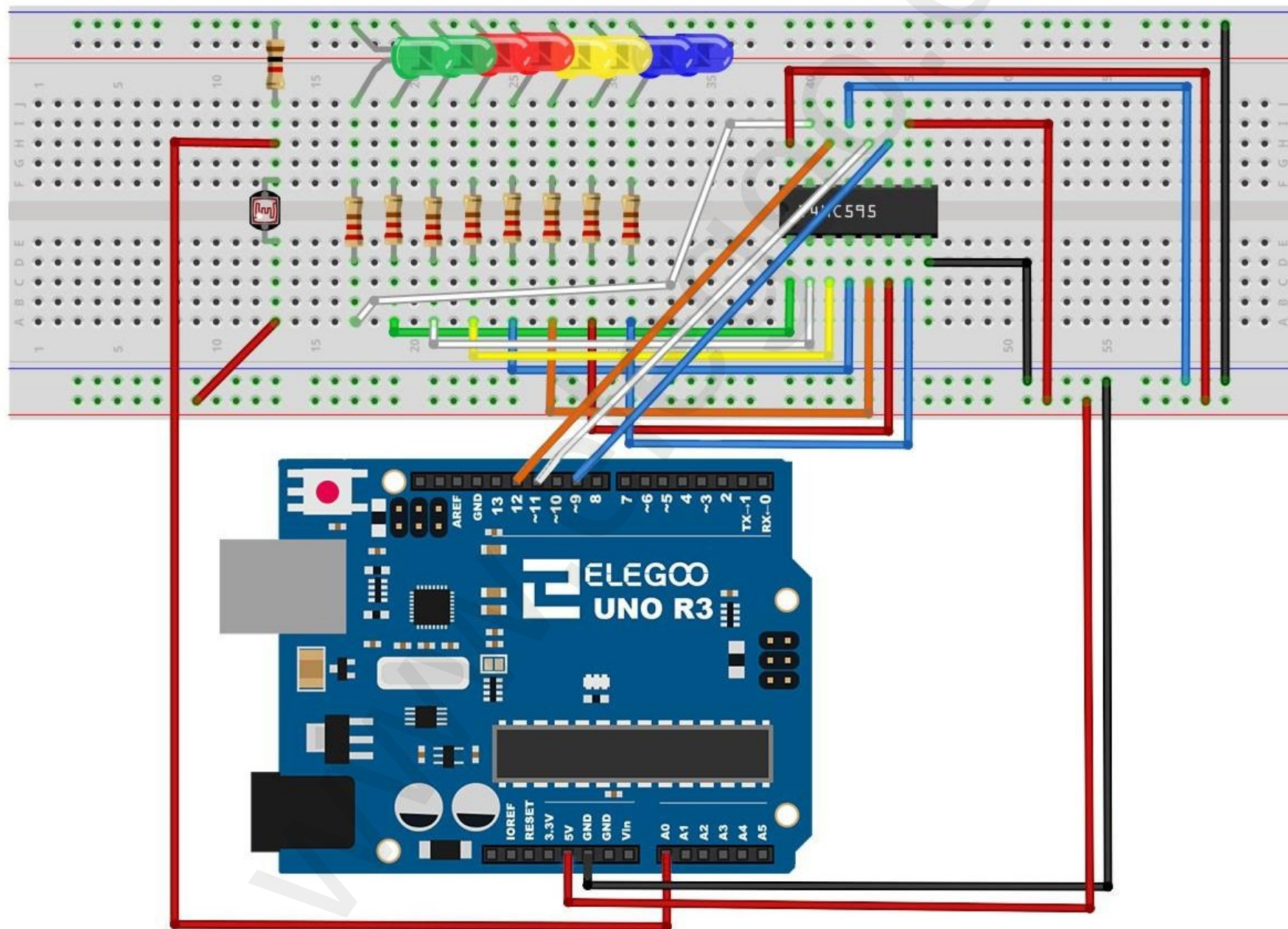


Schéma zapojení



Kód

Po zapojení otevřete program ve složce s kódem - Lekce 26 Fotobuňka a kliknutím na tlačítko **UPLOAD** nahrajte program. Podrobnosti o nahrávání programu naleznete v Lekci 2, pokud se vyskytnou nějaké chyby.

Nejdříve je třeba si všimnout, že jsme změnili název analogového pinu na "lightPin", a nikoli na "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 k Ω . To znamená, že když má fotobuňka odpor 1 k Ω (stejný jako pevný odpor), hrubý údaj bude $1023 / 2 = 511$. To bude odpovídat tomu, že se rozsvítí všechny LED diody, a pak bude bit (numLEDsLit) 8.

Příklad obrázku

