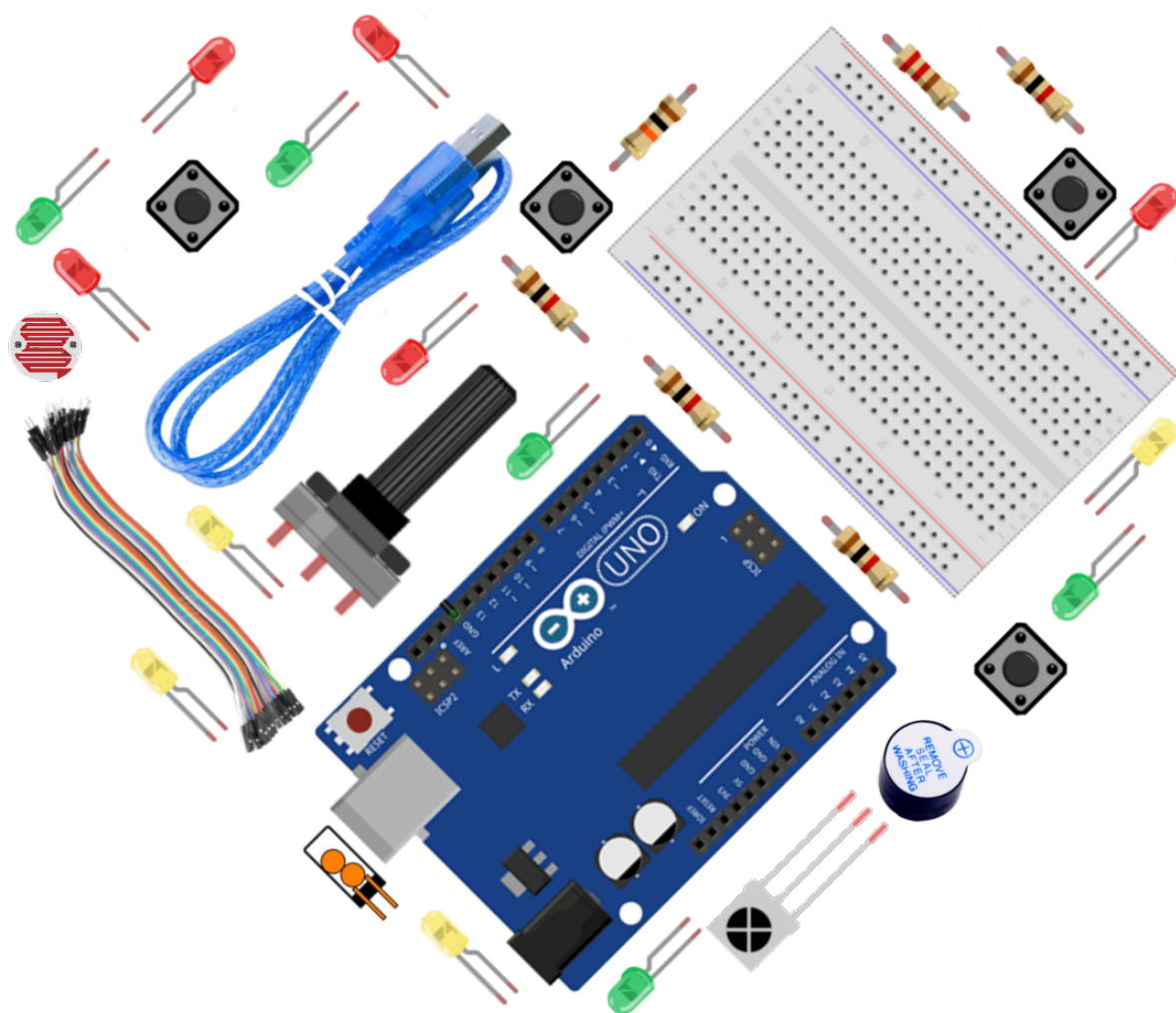
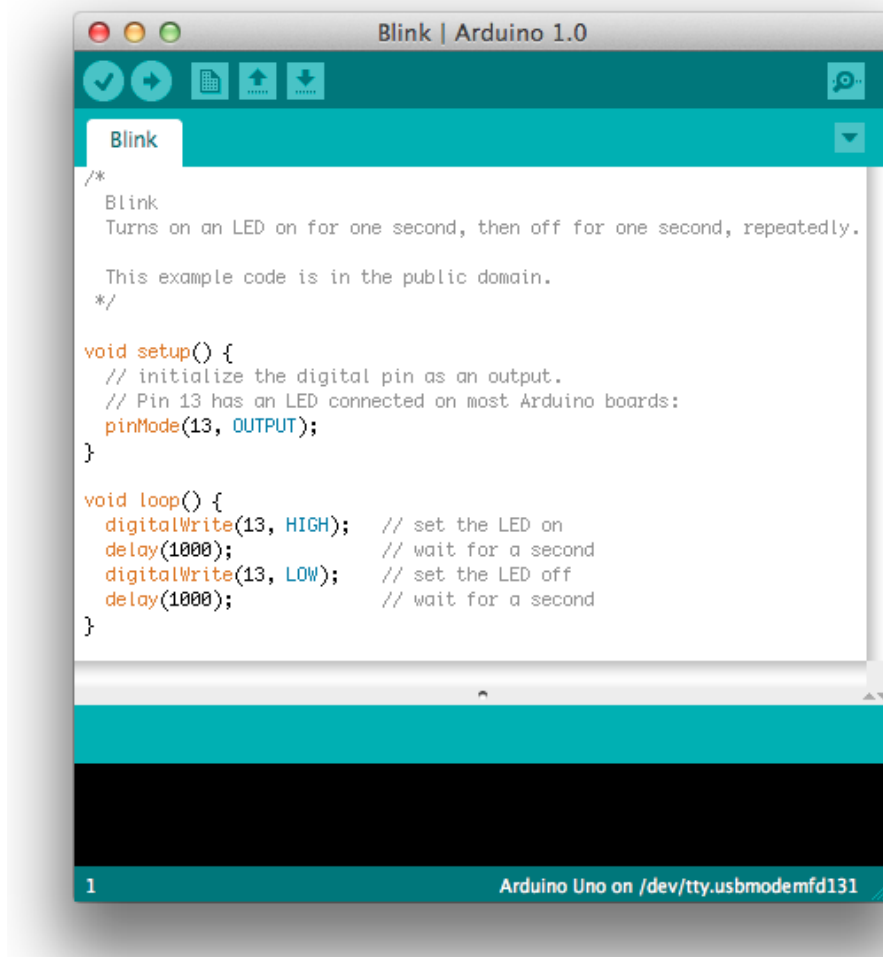




Arduino MINI Starter kit OD BASTLÍŘŮ BASTLÍŘŮM

ARDUINO – ZAČÍNÁME!	2
Lekce 1 Řízení LED tlačítkem	4
Lekce 2 Řízení LED PWMkou	6
Lekce 3 LED Tekoucí potok	7
Lekce 4 Interaktivní tekoucí LED světla	9
Lekce 5 Bzučák	10
Lekce 6 Otřesové čidlo	12
Lekce 7 Kvízový bzučák.....	14
Lekce 8 Sériový monitor	17
Lekce 9 Fotorezistor	19
Lekce 10 Ovládání zvuku pomocí světla	21
Lekce 11 Voltmetr	23
Lekce 12 RGB LED	24
Lekce 13 Infračervený přijímač.....	25

ARDUINO – ZAČÍNÁME!



Tento článek popisuje, jak připojit Tvoje Arduino k počítači a nahrát první projekt (sketch).

1. Připrav si Arduino a USB kabel

V tomto tutoriálu předpokládáme, že používáš Arduino Uno, Mega 2560. Dále budeš potřebovat standardní USB kabel (A plug to B plug).



2. Stáhni si prostředí Arduino

Stáhni si poslední verzi programu z <https://arduino.cc/en/Main/Software>.

Jakmile se stahování dokončí, rozbal stažený soubor. Dej si pozor, abys zachoval strukturu souborů ve složce. Dvojklikem soubor otevři. Ve složce by mělo být několik souborů a podsložek.

3. Zapoj desku

Arduino Uno či Mega mohou být napájeny z USB nebo z externího zdroje. Připoj desku Arduino ke svému počítači použitím USB kabelu. LED na zdroji (označená PWR) by se měla rozsvítit.

4. Nainstaluj ovladače

Instalace ovladačů pro Arduino Uno nebo Arduino Mega 2560 s převodníkem ATmega16U2 (Precizní klon) na Windows 7, Vista a XP (Pro macOS není potřeba).

- Zapoj svoji desku a počkej, až Windows začne instalovat ovladače.
- Klikni na nabídku „Start“ a otevři ovládací panely.
- V nabídce ovládacích panelů klikni na položku „Systém“. Jakmile se Ti otevře nabídka systému, klikni na „Správce zařízení“ (může být potřeba oprávnění správce).
- Rozbal Porty (COM a LPT). Měl bys vidět otevřený port pojmenovaný „Arduino UNO (COMxx)“. Pokud nenajdeš sekci COM a LPT, vyhledej v Ostatních zařízeních „Neznámé zařízení“.
- Pravým tlačítkem klikni na „Arduino UNO (COMx)“ (popřípadě na neznámé zařízení) a vyber možnost „Aktualizovat ovladač“.
- Dále vyber možnost „Prohledat počítač a najít ovladače“.
- Nakonec, najdi a vyber soubor ovladače pojmenovaný „arduino.inf“, který se nachází ve složce „Drivers“ staženého softwaru Arduino.
- Windows dokončí instalaci ovladače.

Instalace ovladačů pro všechny desky Arduino, WeMos atd. s převodníkem CH340 na Windows 7.

Jakmile zapojoš svoji desku do počítače, Windows by měl začít instalovat ovladače (pokud jsi již předtím na počítači nepoužíval desky s CH340).

- Jakmile se zobrazí okno dialogu „Automaticky se spojit s Windows Update a vyhledat software?“, vyber „Tentokrát ne“. Klikni na další.
- Vyber „Nainstalovat ze seznamu“, nebo z umístění (Pokročilé). Klikni na další.
- Ujisti se, že je zaškrtnuta volba „Hledat nejlepší ovladač“ v těchto umístěních; označ volbu „Prohledat výsuvná zařízení“; zaškrtni volbu „Zahrň toto umístění do hledání“ a najdi složku s ovladačem stažených z webových stránek [WeMos](#).
- Panel začne hledat ovladače a pak zobrazí zprávu „USB–SERIAL CH340(COMxx)“ (nebo podobný) byl nalezen. Klikni na „dokončit“.
- Znovu se zobrazí panel „Přidat nové zařízení“. Projdi znovu těmi stejnými kroky a vyber ty samé volby pro umístění a hledání. Tentokrát bude nalezen „USB–SERIAL CH340(COMxx)“ (nebo podobný).

O tom, zda byly ovladače správně nainstalovány, se můžeš ujistit otevřením „Správce zařízení“ (v záložce Hardware panelu Systém). V sekci „Porty“ najdi „USB–SERIAL CH340(COMxx)“ (nebo podobný). To je deska Arduino.

Pro macOS je potřeba stáhnout a nainstalovat ovladače z webových stránek [WeMos](#)!

5. Spust aplikaci Arduino

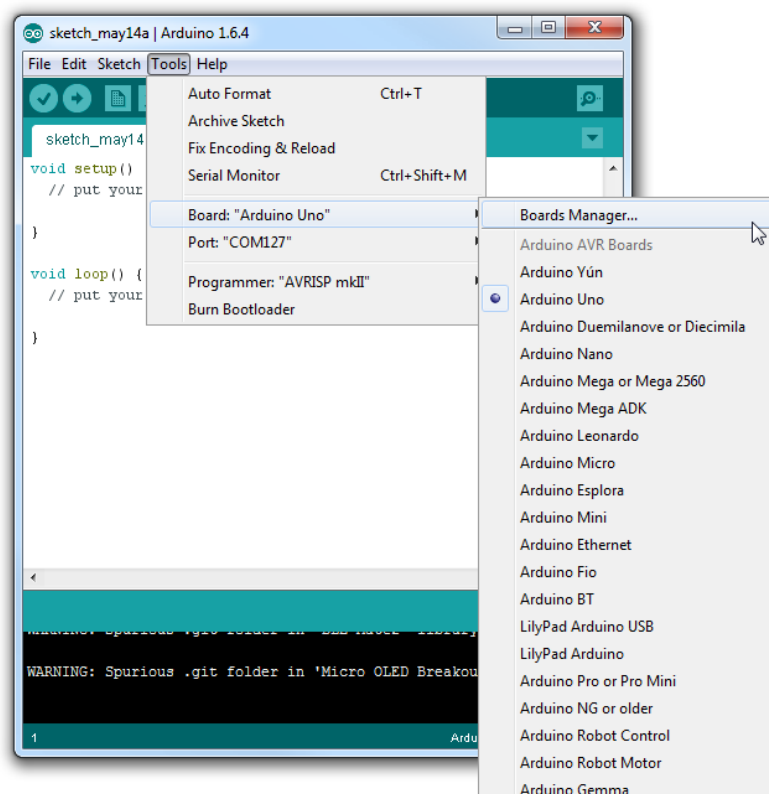
Dvojklikem otevři aplikaci Arduino.

Otevři LED blink example sketch (ukázkový příklad): File > Examples > 1.Basics > Blink.

7. Vyber svoji desku

Vyber v menu Tools > Board (nástroje > deska) druh svého Arduino zařízení.

Vyber v menu Tools > Port (nástroje > port) port svého Arduino zařízení.



9. Nahraj program

Teď jednoduše klikni na tlačítko „Upload“ v prostředí programu. Počkej několik sekund – měl bys vidět blikat RX a TX LED diody na desce. Pokud je nahrávání dokončeno, zobrazí se na status baru zpráva „Done uploading“ - nahrávání dokončeno. Několik sekund potom, co se dokončí nahrávání, bys měl vidět LED piny 13(L) na desce blikat. Pokud blikají, gratulujeme! Tvoje Arduino funguje správně.

Lekce 1 Řízení LED tlačítkem

Úvod

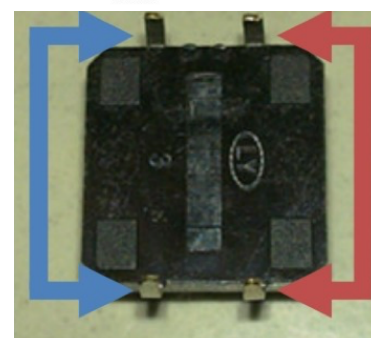
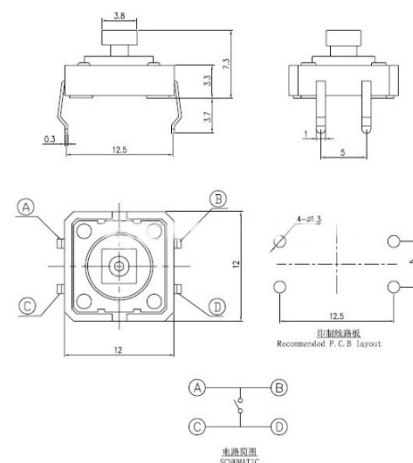
V tomto experimentu se dozvíš, jak zapnout / vypnout LED pomocí I/O portu a tlačítka. "I/O port" odkazuje na výstupní a vstupní port. Zde je použit vstupní port Arduino Uno k tomu, aby přečetl výstup z externího zařízení. Vzhledem k tomu, že samostatná deska je vybavena LED (a připojena do pinu 13), můžeš tuto LED použít pro pohodlnější provedení experimentu.

Komponenty

- Arduino
- USB kabel
- Tlačítko
- Odpor (10kΩ)
- Breadboard vodiče
- Breadboard

Princip

Tlačítka jsou běžnou součástí a slouží k ovládání elektronických zařízení. Jsou obvykle používána jako přepínače pro připojení nebo odpojení obvodů. Tlačítka se vyrábí v



různých tvarech a velikostech, avšak zde jsme použili 12 mm tlačítko, viz následující obrázky. Piny, na které ukazují šipky stejné barvy, jsou propojeny.

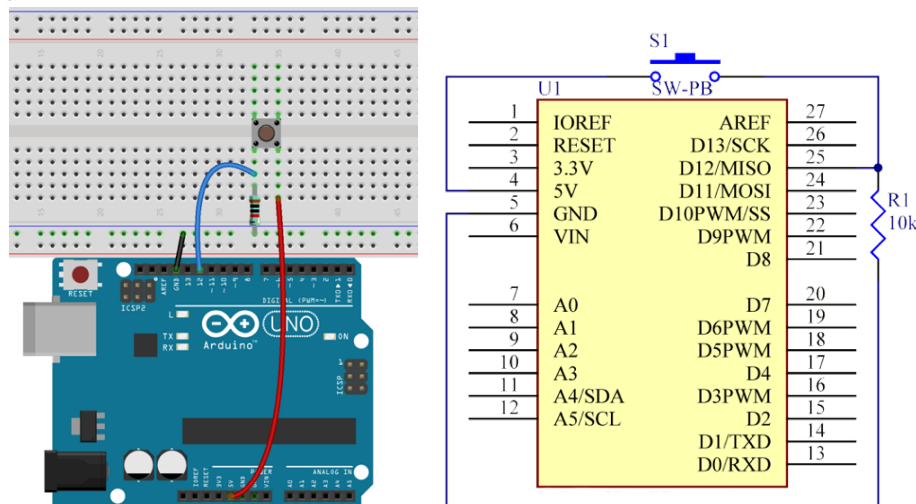
Při stisknutí tlačítka se piny, na které ukazují modré šipky, připojí na piny označené červenou šipkou.

Obecně platí, že tlačítko je přímo napojeno na LED za účelem zapnutí a vypnutí LED. Toto spojení je relativně jednoduché. Někdy se ale LED rozsvítí automaticky, bez stisknutí tlačítka. To může být způsobeno různým typem rušení. Aby se zabránilo těmto vnějším zásahům, je použit pull-down rezistor pro připojení 1K–10KΩ odporu mezi tlačítkem, portem a GND. Toto se využívá k ničení vnějších zásahů, zatímco je připojeno GND tak dlouho, dokud je tlačítko vypnuté.

Toto připojení obvodu je široce používáno v řadě obvodů a elektronických zařízení. Například, pokud stiskneš jakékoli tlačítko na svém telefonu, rozsvítí se Ti podsvícení.

Postup experimentu

Krok 1: Vytvoř obvod



Krok 2: Napiš kód do Arduino IDE

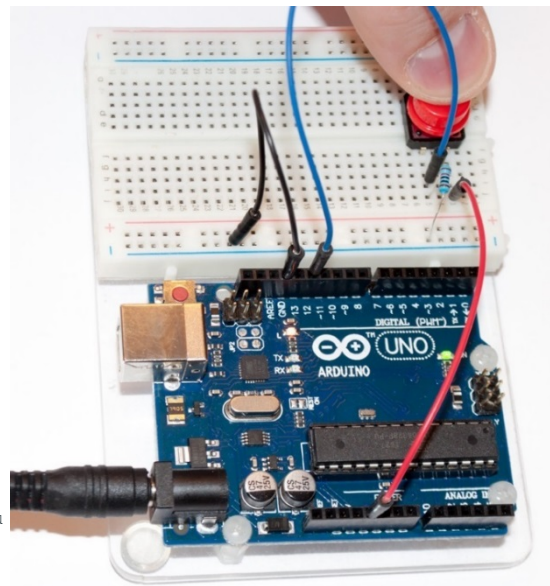
Kód

```
// Řízení LEDky tlačítkem
// Zapíná a vypíná LEDku když stiskneš tlačítko
// Email:laskarduino@gmail.com
// Web:laskarduino.cz
// */

const int keyPin = 12;    // číslo pinu tlačítka
const int ledPin = 13;    // číslo pinu LEDky

void setup() {
    // nastavení pinu tlačítka jako vstupu
    pinMode(keyPin, INPUT);
    // nastavení pinu LEDky jako výstupu
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // přečíst stav hodnoty pinu tlačítka
    // a zkontrolovat, jestli je tlačítko
    // stisknuté
    // pokud ano, stav pinu bude HIGH
    if(digitalRead(keyPin) == HIGH ) {
        digitalWrite(ledPin, HIGH);    // zapnout LEDku
    } else {
        digitalWrite(ledPin, LOW);    // vypnout LEDku
    }
}
```



Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Nyní stiskni tlačítko a LED na Arduino se rozsvítí.

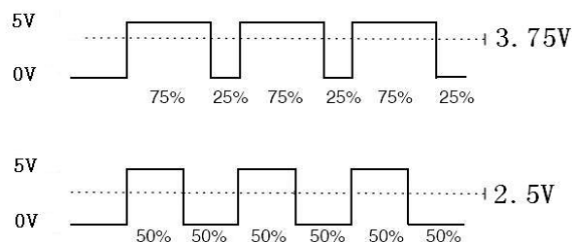
Lekce 2 Řízení LED PWMkou

Úvod

Pojďme zkusit v této lekci něco trochu jednoduššího – postupně budeme měnit jas LEDky pomocí kódů. Vzhledem k tomu, že pulzující světlo vypadá jako dýchání, dali jsme mu magické jméno – dýchající LEDka. Dosáhneme tohoto efektu pomocí pulzně šířkové modulační (PWM)

Komponenty

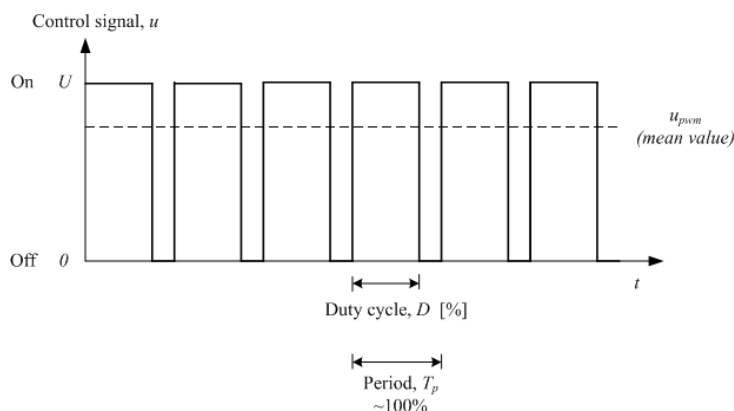
- Arduino
- Breadboard
- Breadboard vodiče
- 1x LED
- Odpor (220Ω)
- USB kabel



Princip

Pulzně šířková modulační neboli PWM (Pulse Width Modulation) je diskretní modulační pro přenos analogového signálu pomocí dvouhodnotového signálu. Jako dvouhodnotová veličina může být použito např. napětí. Signál je přenášen pomocí střídavy. Vzhledem ke svým vlastnostem je pulzně šířková modulační často využívána ve výkonové elektronice pro řízení velikosti napětí.

Přenosový signál, který nese informaci o přenášené hodnotě, může nabývat hodnot zapnuto/vypnuto tj. log.1/log.0 (5V a 0V v našem případě). Hodnota přenášeného signálu je v přenosu "zakódována" jako poměr mezi stavy zapnuto/vypnuto. Tomuto poměru se říká střída. Cyklu, kdy dojde k přenosu jedné střídavy, se říká perioda. Omezením pro PWM je to, že přenos informace je vždy omezen na relativní vyjádření a to 0 - 100 %. To znamená, že musí být znám poměr mezi skutečnou hodnotou a procentuálním vyjádřením. Časové hodnoty střídavy se pohybují v sekundách, v milisekundách pro přesnější řízení. Perioda je vždy součtem doby zapnuto a vypnuto. Musíš zopakovat periodu dostatečně rychle s LED, aby nebylo vidět, že LEDka bliká, ale trvale svítí.



Z oscilogramu je patrné, že amplituda napětí výstupu je 5V. Skutečné výstupní napětí je pouze 3.75V pomocí PWM, protože 5V trvá pouze 75% střídavy (duty cycle).

Tři základní parametry PWM:

1. Střída (duty cycle) je poměr mezi stavy zapnuto/vypnuto.
2. Perioda je součtem doby zapnuto a vypnuto.
3. Amplituda napětí je zde 0V-5V.

Postup experimentu

Krok 1: Vytvoř obvod



The image shows an Arduino Uno R3 microcontroller board. It is connected to a white breadboard. A red LED is connected to the breadboard, with its anode (longer leg) connected to a red wire that goes to the 5V pin on the Arduino. The cathode (shorter leg) is connected to a black wire that goes to the GND pin. A black USB Type-A to micro-USB cable is plugged into the USB port on the left side of the board. The board is labeled 'ARDUINO UNO' and 'DIGITAL (POWER)'.

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Tady bys měl vidět, jak se LEDka stává jasnější a tmavší, a tak pořád dokola.

Lekce 3 LED Tekoucí potok

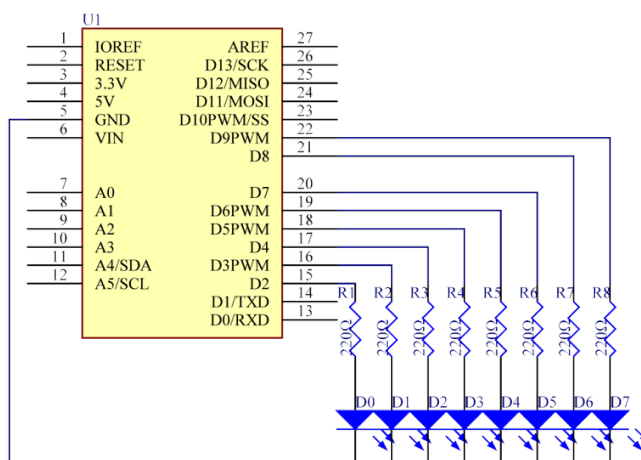
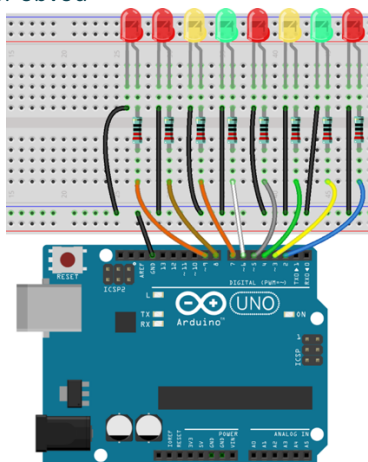
V této lekci uděláme jednoduchý, ale zajímavý experiment – pomocí LED vytvořím tekoucí LED světla. Jak název napovídá, tato plynulá světla jsou tvořena osmi LED v řadě, která se postupně rozsvítí a tmavnou jeden po druhém, stejně jako tekoucí voda.

Komponenty

- Arduino
- Breadboard
- Breadboard vodiče
- 8x LED
- 8x Otpor (220Ω)
- USB kabel

Princip tohoto experimentu je jednoduchý – zapnout osm LED za sebou.

Krok 1: Vytvoř obvod

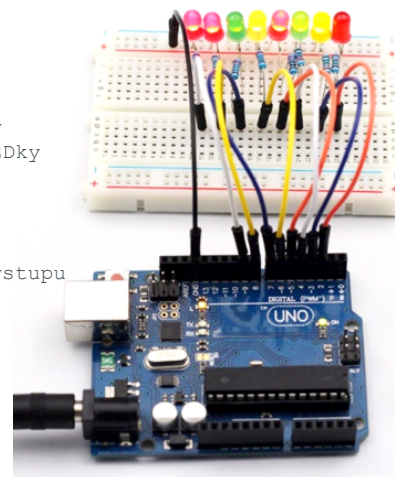


Kód

```
const int lowestPin = 2;           // číslo pinu první LEDky
const int highestPin = 9;         // číslo pinu poslední LEDky

void setup() {
    // nastavení pinu od 2 do 9 LEDky jako výstupu
    for(int thisPin = lowestPin; thisPin <= highestPin; thisPin++) {
        pinMode(thisPin, OUTPUT);    // nastavení pinu jako výstupu
    }
}

void loop() {
    // iterace pinu za pinem
    // zapnout LEDky od první do poslední
    for(int thisPin = lowestPin; thisPin <= highestPin; thisPin++) {
        digitalWrite(thisPin, HIGH); // zapnout tuto LEDku
        delay(100);                  // počkat 100 ms
    }
    // vypnout LEDky od poslední do první
    for(int thisPin = highestPin; thisPin >= lowestPin; thisPin--) {
        digitalWrite(thisPin, LOW);  // vypnout tuto LEDku
        delay(100);                  // počkat 100 ms
    }
    for(int thisPin = highestPin; thisPin >= lowestPin; thisPin--) {
        digitalWrite(thisPin, HIGH); // zapnout tuto LEDku
        delay(100);                  // počkat 100 ms
    }
    for(int thisPin = lowestPin; thisPin <= highestPin; thisPin++) {
        digitalWrite(thisPin, LOW);  // vypnout tuto LEDku
        delay(100);                  // počkat 100 ms
    }
}
```



Nyní bys měl vidět osm LED světél, která se rozzáří jedno po druhém - zleva doprava, a potom ztlumeně postupně - zprava doleva. A zase opačně. Celý tento proces se bude opakovat, dokud bude obvod napájen.

Lekce 4 Interaktivní tekoucí LED světla

Úvod

V minulé lekci ses už naučil, jak se dělají tekoucí LED světla. V této lekci přidáme potenciometr, který bude měnit interval blikání pomocí nastavení potenciometru.

Komponenty

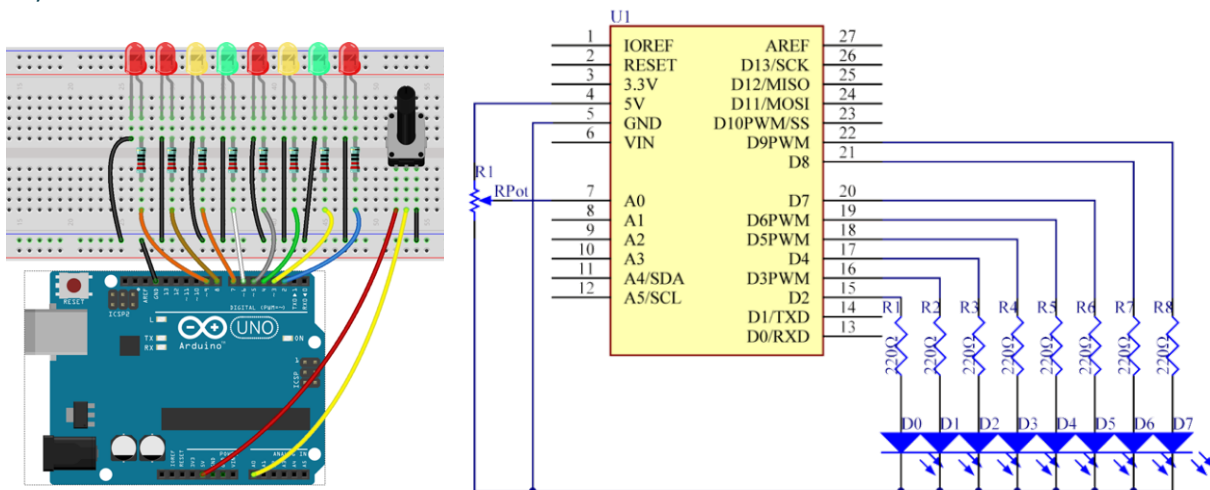
- Arduino
- Breadboard
- 8x LED
- 8x Odpor (220Ω)
- Potentiometer
- USB kabel
- Breadboard vodiče

Princip

Princip je poměrně jednoduchý. Zapne se 8 LEDek podle pořadí. A pak se změní časový interval zapnutí a vypnutí LED pomocí nastavení potenciometru.

Postup experimentu

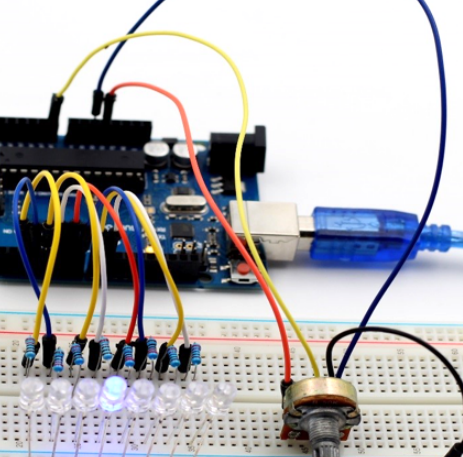
Krok 1: Vytvoř obvod



Krok 2: Napiš kód do Arduino IDE

Kód

žrečíst hodnotu pinu potenciometru



Princip

Jedná se o elektrický bzučák s integrovanou strukturou, a proto jsou bzučáky, jež jsou napájeny stejnosměrným napětím, hojně využívány v počítačích, tiskárnách, kopírkách, alarmech, elektrických hračkách, automobilových elektronických zařízeních, telefonech, časovačích a jiných elektronických výrobcích s hlasovým zařízením. Bzučáky můžeme rozdělit na aktivní nebo pasivní (viz následující obrázek). Otoč kuličky obou bzučáků lícem nahoru - ten se zelenou kulatou destičkou je pasivní, zatímco ten zalitý černým lepidlem (polymerem) je bzučák aktivní.



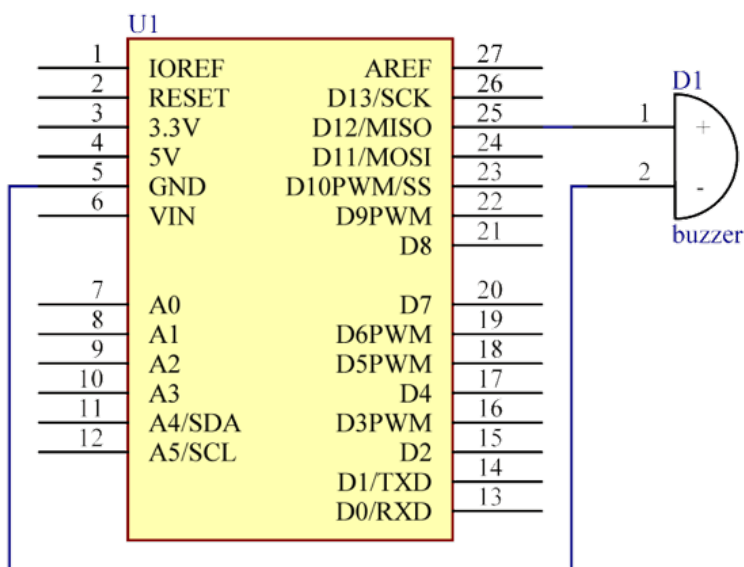
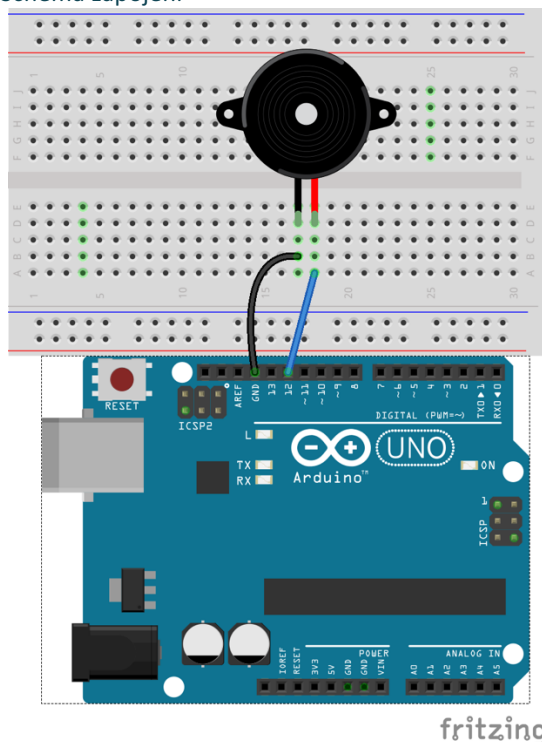
Rozdíl mezi aktivním a pasivním bzučákem:

Aktivní bzučák má v sobě zabudovaný oscilační zdroj, takže bude vydávat zvuky, pokud bude pod proudem. Pasivní bzučák ale takový zdroj nemá, proto při použití stejnosměrného napětí pípat nebude – místo toho je potřeba použít čtvercové vlny s frekvencí mezi 2K a 5K. Aktivní bzučák bývá často nákladnější a to právě z důvodu několika zabudovaných oscilátorů. V následujícím pokusu použijeme aktivní bzučák.

Postup experimentu

Krok 1: Vytvoř obvod

Schéma zapojení



Krok 2: Napiš kód do Arduino IDE

Kód

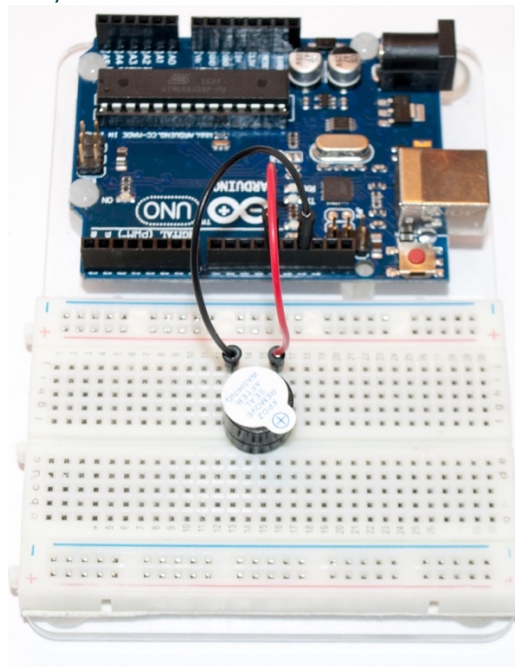
```
// Bzučák  
// Bzučák můžeš využít kdykoli, pokud budeš chtít použít nějaký zvuk.  
// Email:laskarduino@gmail.com  
// Web:laskarduino.cz  
/*\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ */  
  
int buzzer = 12; // číslo pinu bzučáku  
void setup() {  
    pinMode(buzzer, OUTPUT); // nastavení pinu bzučáku jako výstupu
```

```
}  
void loop() {  
    unsigned char i;                // definuj proměnnou  
    while(1) {  
        // generace frekvence  
        for(i=0;i<80;i++) {  
            digitalWrite(buzzer,HIGH);  
            delay(1);                // počkat 1ms  
            digitalWrite(buzzer,LOW);  
            delay(1);                // počkat 1ms  
        }  
        //generace jiné frekvence  
        for(i=0;i<100;i++) {  
            digitalWrite(buzzer,HIGH);  
            delay(2);                // počkat 2ms  
            digitalWrite(buzzer,LOW);  
            delay(2);                // počkat 2ms  
        }  
    }  
}
```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Nyní bys měl slyšet, jak bzučák vytváří zvuky.



Lekce 6 Otřesové čidlo

Úvod

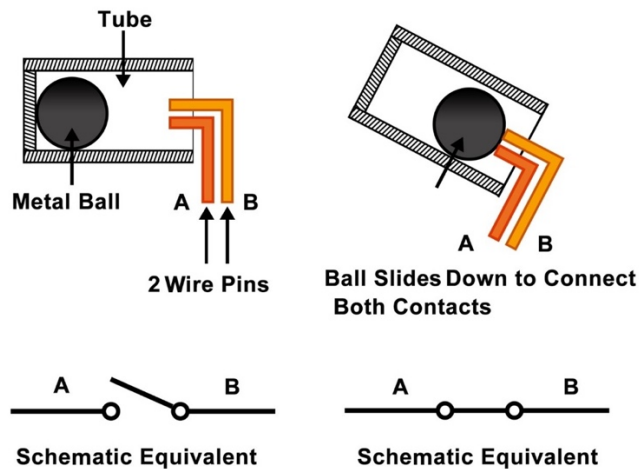
Jde o kulový spínač s kovovou kuličkou uvnitř. Používá se k detekci malého úhlu sklonu.

Komponenty

- Arduino
- USB kabel
- Otřesové čidlo
- Breadboard vodiče

Princip

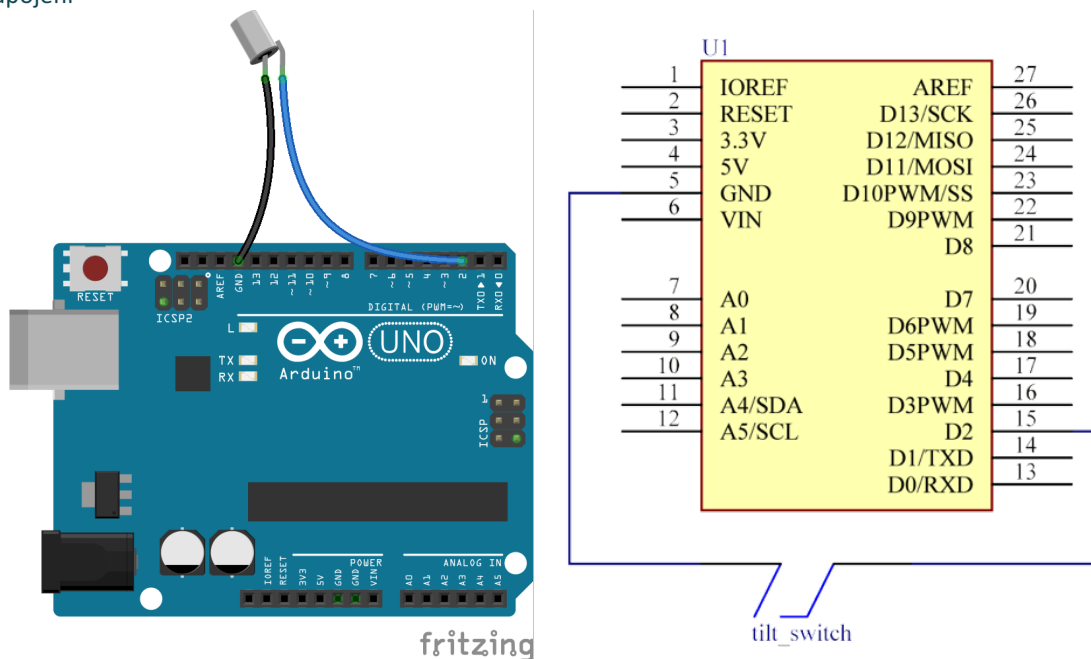
Princip je velice jednoduchý. Je-li spínač nakloněn v určitém úhlu, kulička uvnitř sjede dolů a dotkne se dvou kontaktů připojených ke vnějším kolíčkům. Tím propojí kontakty. Pokud zůstane kulička mimo kontakty, rozpojí je.



Postup experimentu

Krok 1: Vytvoř obvod

Schéma zapojení



Krok 2: Napiš kód do Arduino IDE

Kód

```
// Otřesové čidlo
// Zapíná a vypíná LEDku když nakloníš čidlo.
// Email:laskarduino@gmail.com
// Web:laskarduino.cz
// ////////////////////////////////////////////

const int ledPin = 13; // číslo pinu LEDky
const int tiltPin = 2; // číslo pinu otřesového čidla

void setup() {
    pinMode(ledPin, OUTPUT); // nastavení pinu LEDky jako výstupu
    pinMode(tiltPin, INPUT); // nastavení pinu otřesového čidla jako vstupu
}

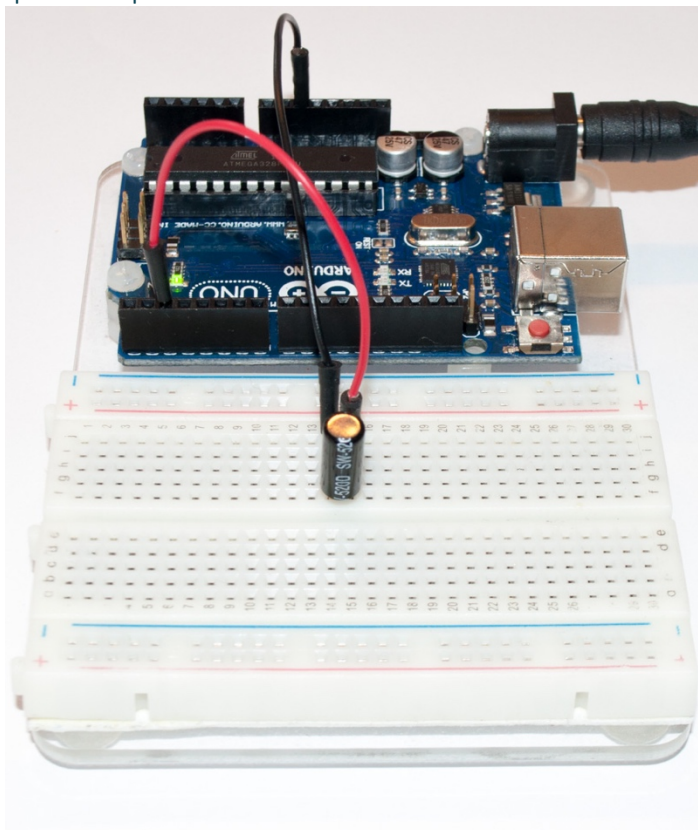
void loop() {
    int digitalVal = digitalRead(tiltPin); // přečíst stav hodnoty pinu otřesového čidla
```

```
    delay(300); // počkat 300ms
    if(HIGH == digitalVal) { // pokud není otřesové čidlo nakloněno
        digitalWrite(ledPin, LOW); // vypnout LEDku
    } else { // pokud je otřesové čidlo nakloněno
        digitalWrite(ledPin, HIGH); // zapnout LEDku
    }
}
```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Pohněte čidlem a LED, jež je připevněna k pinu 13 na Arduino. Rozsvítí se.



Lekce 7 Kvízový bzučák

Úvod

Ve vědomostních soutěžích, především těch zábavních (např. soutěže v odpovídání na otázky), organizátoři často používají bzučákový systém k přesnému a spravedlivému určení čísla odpovídajícího. Nyní je systém schopný objektivně ilustrovat přesnost a spravedlivost soutěže, čímž dělá pořad zábavnější. V této lekci budeme používat tlačítka, bzučáky a LED a vytvoříme kvízový bzučák.

Komponenty

- Arduino
- USB kabel
- 4 Tlačítka
- 4x LED
- 4x Odpor (220Ω)
- Aktivní bzučák
- Breadboard vodiče
- Breadboard


```
#define button1 2          // číslo pinu 1 tlačítka
#define button2 3          // číslo pinu 2 tlačítka
#define button3 4          // číslo pinu 3 tlačítka
#define button4 9          // číslo pinu 4 tlačítka
#define buzzerPin 5        // číslo pinu bzučáku
#define LED1 6             // číslo pinu 1 LEDky
#define LED2 7             // číslo pinu 2 LEDky
#define LED3 8             // číslo pinu 3 LEDky
#define LED4 10            // číslo pinu 4 LEDky
#define uint8 unsigned char
uint8 flag = 0;           // indikace stavu tlačítka 4
uint8 b1State,b2State,b3State,b4State = 0;

void setup() {
    // nastavení pinů LEDek a bzučáku jako výstupu
    pinMode(buzzerPin, OUTPUT);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(LED4, OUTPUT);
    // nastavení pinů tlačítek jako vstupu s pullup odporem
    pinMode(button1, INPUT_PULLUP);
    pinMode(button2, INPUT_PULLUP);
    pinMode(button3, INPUT_PULLUP);
    pinMode(button4, INPUT_PULLUP);
    // vypnout všechny LEDky
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
}

void loop() {
    // vypnout všechny LEDky
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    // přečíst stav pinu tlačítka 4
    b4State = digitalRead(button4);
    // pokud je tlačítko 4 stisknuté
    if(b4State == 0) {
        flag = 1; // nastavím flag na 1
        digitalWrite(LED4, HIGH); // zapnout 4 LEDku
        delay(200);
    }
    if(1 == flag) {
        // přečíst stav pinů tlačítek 1 až 3
        b1State = digitalRead(button1);
        b2State = digitalRead(button2);
        b3State = digitalRead(button3);
        // pokud bylo tlačítko 1 stisknuto jako první
        if(b1State == 0) {
            flag = 0;
            digitalWrite(LED4, LOW);
            Alarm(); // zvuk bzučáku
            digitalWrite(LED1,HIGH); // zapnout jen 1 LEDku
            digitalWrite(LED2,LOW);
            digitalWrite(LED3,LOW);
            // čekání na stisknutí tlačítka 4
            while(digitalRead(button4));
        }
        // pokud bylo tlačítko 2 stisknuto jako první
        if(b2State == 0) {
            flag = 0;
            digitalWrite(LED4, LOW);
            Alarm();
            digitalWrite(LED1,LOW);
            digitalWrite(LED2,HIGH);
            digitalWrite(LED3,LOW);
            while(digitalRead(button4));
        }
        // pokud bylo tlačítko 3 stisknuto jako první
        if(b3State == 0) {
            flag = 0;
        }
    }
}
```

```

        digitalWrite(LED4, LOW);
        Alarm();
        digitalWrite(LED1, LOW);
        digitalWrite(LED2, LOW);
        digitalWrite(LED3, HIGH);
        while(digitalRead(button4));
    }
}

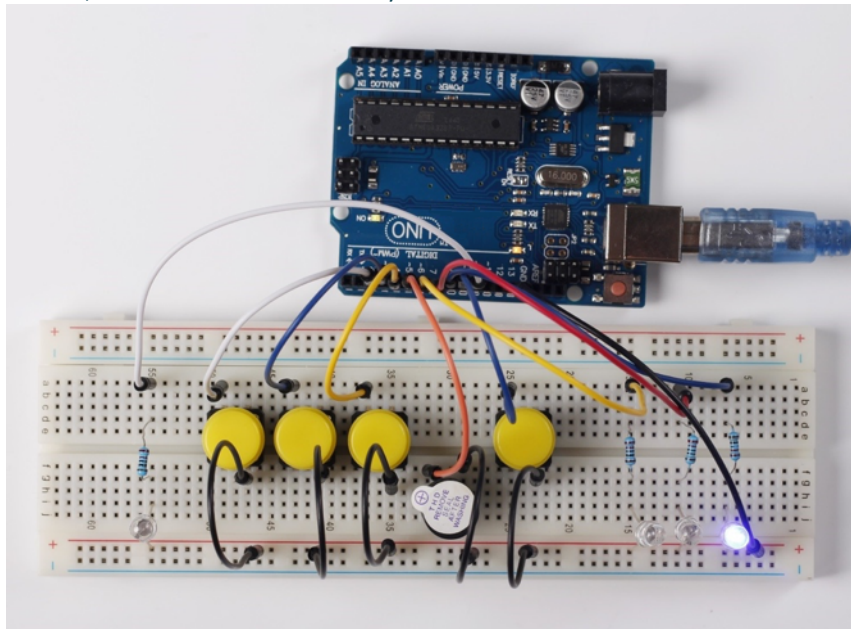
// zvuk bzučáku
void Alarm() {
    for(int i=0; i<300; i++) {
        digitalWrite(buzzerPin, HIGH);    // zapnout bzučák
        delay(1);                          // pauza nastaví frekvenci
        digitalWrite(buzzerPin, LOW);     // vypnout bzučák
        delay(1);                          // pauza nastaví frekvenci
    }
}

```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Nyní nejprve zmáčkní tlačítko 4. Pokud prvně zmáčkneš tlačítko 1, rozsvítí se odpovídající LED a pípne bzučák. Poté opět zmáčkní tlačítko 4 k resetování, než začneš s dalšími tlačítky.



Lekce 8 Sériový monitor

Úvod

V této lekci se naučíme, jak zapnout a vypnout LED pomocí počítače a seriového monitoru. Sériový monitor se používá ke komunikaci mezi Arduino deskou a počítačem nebo jinými přístroji. Jedná se o zabudovaný software v Arduino prostředí a otevře se kliknutím na tlačítko v pravém horním rohu. Můžeš posílat a získávat data pomocí sériového portu na Arduino desce a ovládat desku pomocí klávesnice. V této lekci používáme 3 LED, a proto můžeš vložit červenou, zelenou a žlutou barvu na sériový monitor v IDE. Odpovídající LED na Arduino se tímto rozsvítí.

Komponenty

- Arduino
- Breadboard
- 3x LED
- 3x Odpor (220Ω)

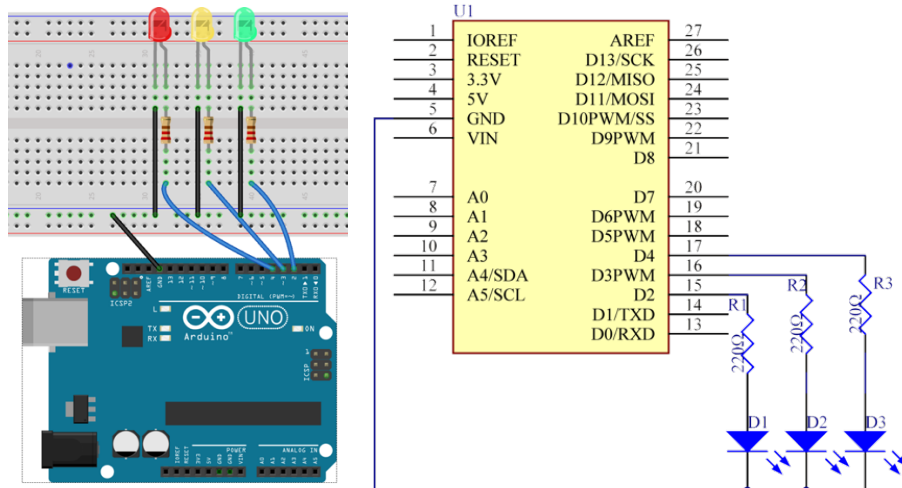
- Breadboard vodiče
- USB kabel

Princip

Zde slouží sériový monitor jako přestupní stanice pro komunikaci mezi tvým počítačem a Arduino. Nejprve převede počítač data do sériového monitoru a ta jsou poté přečtena pomocí Arduino Uno (nebo Mega). Poté provede Uno další operace.

Postup experimentu

Krok 1: Vytvoř obvod



Krok 2: Napiš kód do Arduino IDE

Kód

[illegible]

```

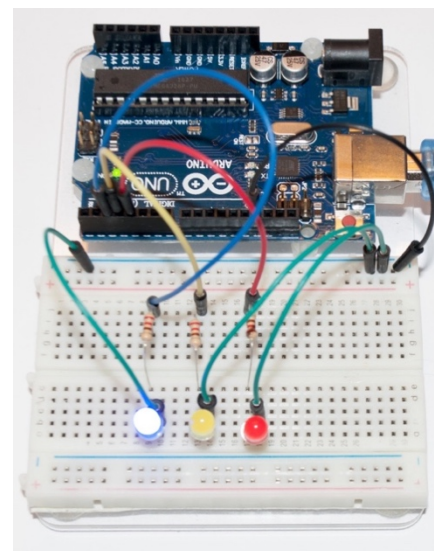
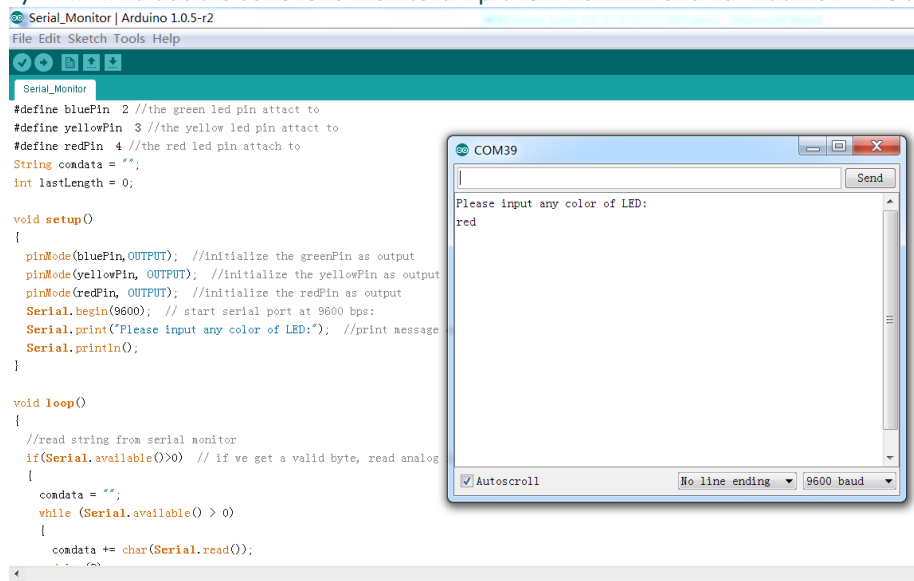
    } else if (comdata == "modra") {
        digitalWrite(cervenaPin, LOW); // vypnout červenou LEDku
        digitalWrite(modraPin, HIGH); // zapnout modrou LEDku
        digitalWrite(zlutaPin, LOW); // vypnout žlutou LEDku
    } else {
        digitalWrite(cervenaPin, LOW); // vypnout červenou LEDku
        digitalWrite(modraPin, LOW); // vypnout modrou LEDku
        digitalWrite(zlutaPin, LOW); // vypnout žlutou LEDku
    }
}

```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Nyní klikni na tlačítko sériového monitoru v pravém horním rohu na Arduino IDE. Objeví se okénko sériového monitoru.



Pomocí tohoto okénka můžeš posílat data z počítače do Arduino, ale také přijímat data a zobrazit si je na obrazovce. Když okno otevřeš, objeví se "Vlož prosím jakoukoli LED barvu." Zde můžeš vložit barvu. Když vložíš červenou, zelenou nebo žlutou, klikni na Poslat a odpovídající LED se rozsvítí. Pokud ale přidáš jinou barvu, nerozsvítí se žádná LED. Například, pokud přidáš červenou, rozsvítí se červené LED.

Lekce 9 Fotorezistor

Úvod

Fotorezistor či photocell je světlem ovládaný proměnný odpor. Odpor fotorezistoru se snižuje s rostoucí intenzitou dopadajícího světla; jinými slovy, je fotovodivý. Fotorezistor můžeme využít pro na světlo citlivé detektory, jež se aktivují světlem či tmou.

Komponenty

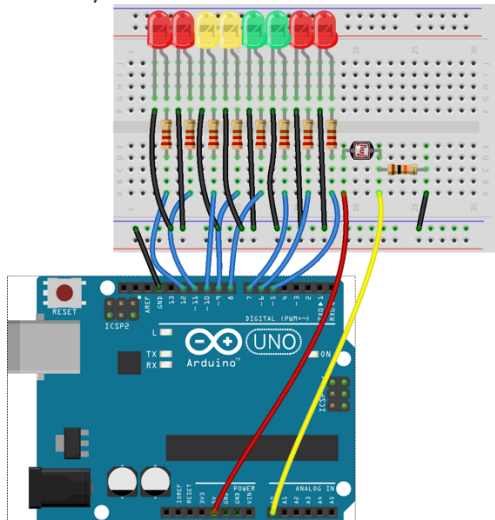
- Arduino
- USB kabel
- Fotorezistor
- Odpor (10KΩ)
- 8x LED
- 8x Odpor (220Ω)
- Breadboard vodiče
- Breadboard

Princip

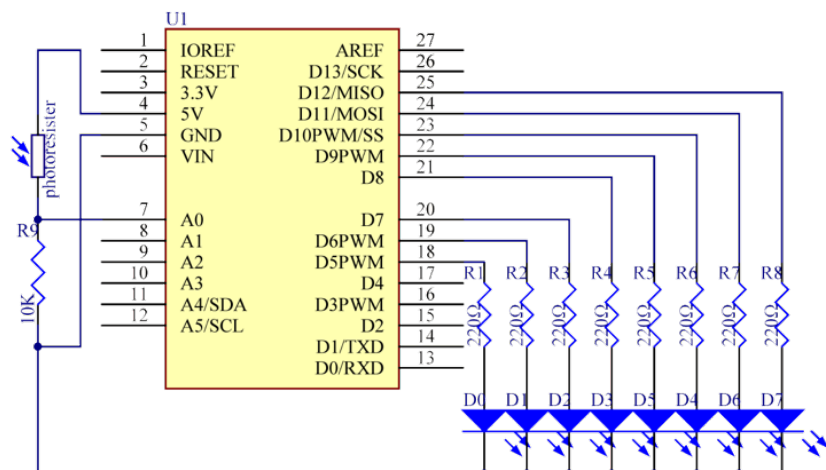
Odpor ve fotorezistoru se mění na základě intenzity dopadajícího světla. Když se intenzita tohoto světla zvýší, odpor se sníží a naopak. V tomto experimentu použijeme 8 LED. Čím větší bude intenzita světla, tím více LED se rozsvítí. Až dosáhneme určité intenzity světla, rozsvítí se všechny LED. V případě žádného světla se nerozsvítí nic.

Postup experimentu

Krok 1: Vytvoř obvod



fritzing



Krok 2: Napiš kód do Arduino IDE

Kód

```
// Fotorezistor
// Pokud bude na fotorezistor svítit světlo určité intenzity, uvidíš určitý počet svítících LEDek.
// Pokud tuto intenzitu zvětšíš, uvidíš víc svítících LEDek.
// Pokud to dáš do tmy, žádná LEDka svítit nebude.
// Email:laskarduino@gmail.com
// Web:laskarduino.cz
// */

const int NbrLEDs = 8; // počet LEDek
const int ledPins[] = {5, 6, 7, 8, 9, 10, 11, 12}; // vytvoříme pole pro LED piny
const int photocellPin = A0; // číslo pinu fotorezistoru

int sensorValue = 0; // přečteny hodnoty z čidla
int ledLevel = 0; // hodnota čidla převedená do LED řádku

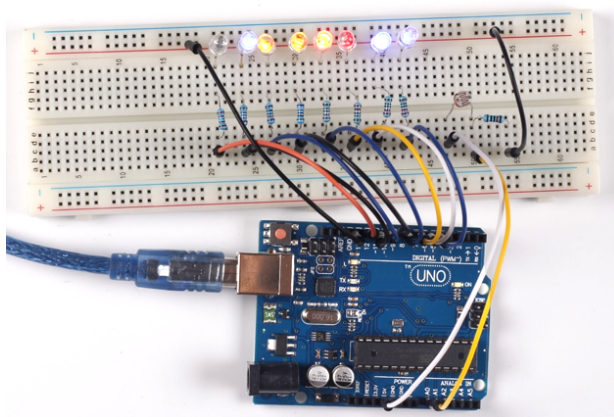
void setup() {
  for (int led = 0; led < NbrLEDs; led++) {
    pinMode(ledPins[led], OUTPUT); // nastavím všechny LED piny jako výstup
  }
}

void loop() {
  sensorValue = analogRead(photocellPin); // přečíst hodnotu pinu A0
  ledLevel = map(sensorValue, 300, 1023, 0, NbrLEDs); // převedeme hodnotu čidla do LED řádku
  for (int led = 0; led < NbrLEDs; led++) { // půjdeme po každé LEDce
    if (led < ledLevel) { // Jestli je číslo LED méně, než ledLevel
      digitalWrite(ledPins[led], HIGH); // zapne tuto LEDku
    } else {
      digitalWrite(ledPins[led], LOW); // jinak vypne tuto LEDku
    }
  }
}
```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Posviť na fotorezistor a uvidíš několik LED, jak se rozsvítí. Posviť více a rozsvítí se LED o to víc. Jakmile fotorezistor postavíš do tmy, všechna LED zhasnou.



Vyzkoušej!

Kromě výše uvedeného experimentu můžeš také vyměnit fotorezistor za mikrofon a pozorovat, jak LED signalizuje sílu zvuku. Čím vyšší bude síla zvuku, tím více LED se rozsvítí. Vyzkoušej si tento pokus na vlastní kůži a uvidíš.

Lekce 10 Ovládání zvuku pomocí světla

Úvod

Už ses naučil, jak pracovat s fotorezistorem a nyní se dozvíš, jak ovládat bzučák pomocí fotorezistoru tak, aby pípal v různých frekvencích.

Komponenty

- Arduino
- USB kabel
- Fotorezistor
- Aktivní bzučák
- 1x Odpor (10K Ω)
- Breadboard vodiče
- Breadboard

Princip

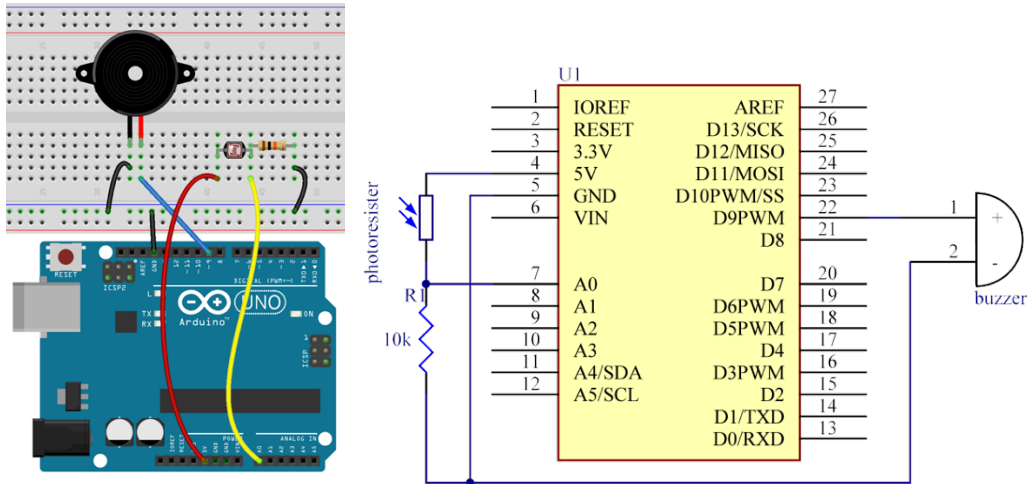
Když nasvítíš fotorezistor a intenzita dopadajícího světla bude vyšší, odpor fotorezistoru se sníží; a naopak.

V tomto pokusu je výstup fotorezistoru vyslán do pinu A0 na Arduino a poté zpracován ADC na desce, kde vytvoří digitální signál. Tento signál používáme jako parametr funkce delay() v kódu, než se ozve bzučák. Pokud je dopadající světlo silné, je výsledná hodnota vyšší, což znamená, že bude bzučák pípat pomalu; je-li světlo slabé, výsledná hodnota je nižší a bzučák bude pípat rychleji.

Postup experimentu

Krok 1: Vytvoř obvod

Schéma zapojení



Krok 2: Napiš kód do Arduino IDE

Kód

```
// Ovládání zvuku pomocí světla
// Pokud dáš fotorezistor do tmavého prostředí, bzučák bude intenzivně pípat.
// Pokud dáš fotorezistor do světlého prostředí, bzučák bude pípat pomalu.
// Email:laskarduino@gmail.com
// Web:laskarduino.cz
// */

const int photocellPin = A0; // číslo pinu fotorezistoru
int sensorValue = 0; // přečteny hodnoty z čidla
const int buzzerPin = 9; // číslo pinu bzučáku

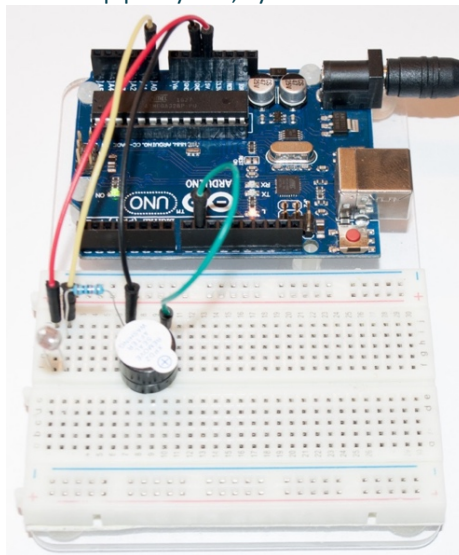
void setup() {
    pinMode(buzzerPin, OUTPUT); // nastavení pinů bzučáku jako výstupu
}

void loop() {
    sensorValue = analogRead(photocellPin); // přečíst hodnotu pinu A0
    digitalWrite(buzzerPin, HIGH); // zapnout bzučák
    delay(sensorValue); // pauza nastaví frekvenci
    digitalWrite(buzzerPin, LOW); // vypnout bzučák
    delay(sensorValue); // pauza nastaví frekvenci
}
```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Pokud postavíte fotorezistor do tmy, bude bzučák pítat rychle; vystavíte-li fotorezistor světlu, bude pítat pomalu.



Lekce 11 Voltmetr

Úvod

V této lekci vytvoříme voltmetr s pomocí potenciometru a sériového monitoru.

Komponenty

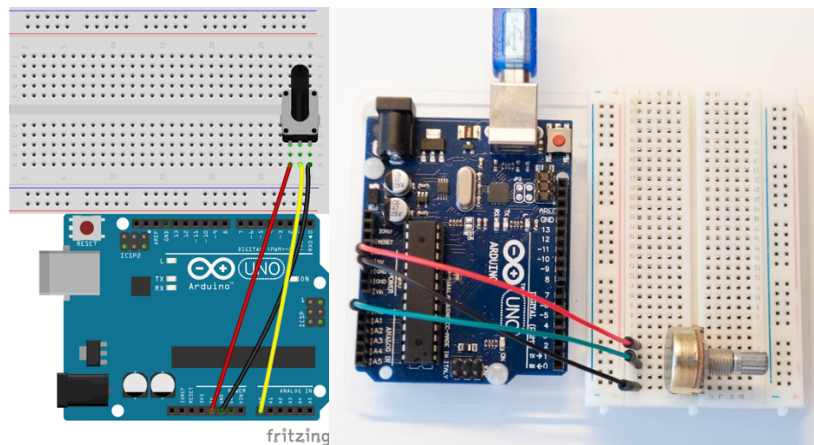
- Arduino
- USB kabel
- Potenciometr
- Breadboard vodiče
- Breadboard

Princip

V tomto experimentu využijeme potenciometr k dělení napětí. Vzhledem k tomu, že Arduino umí číst pouze digitální signály, ale na výstupech posuvných pinů potenciometru je analogový signál, musíme převést tento analogový signál do digitálního s analogově-digitálním konvertorem (ADC). Naštěstí Arduino samo o sobě je dodáváno s 10-bitovým ADC, které můžeme použít k provedení této konverze.

Postup experimentu

Krok 1: Vytvoř obvod



Krok 2: Napiš kód do Arduino IDE

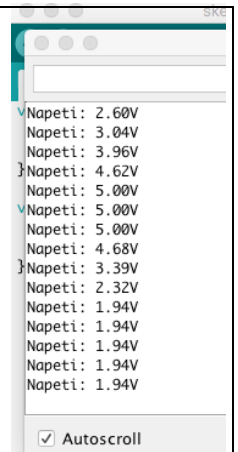
Kód

```
// Voltmetr
// Nastav potenciometr a uvidíš, že napětí zobrazené na sériovém
// monitoru se změnilo odpovídajícím způsobem.
// Email:laskarduino@gmail.com
// Web:laskarduino.cz
// */*****//

float val = 0;
int volt = A0; // číslo pinu pro měření napětí

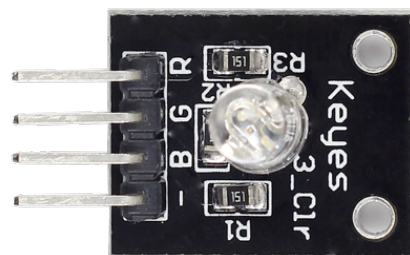
void setup() {
    Serial.begin(9600); // spustit sériový monitor na 9600 bps
}

void loop() {
    val = analogRead(volt); // přečíst hodnotu pinu A0
    val = val/1024*5.0; // konvertování hodnoty pinu do napětí
    Serial.print("Napeti: "); // tisknout "Napětí: " do sériového monitoru
    Serial.print(val); // tisknout napětí do sériového monitoru
    Serial.println("V"); // tisknout "V" do sériového monitoru a přejít na nový řádek (ln)
    delay(300); // počkat 300 ms
}
```

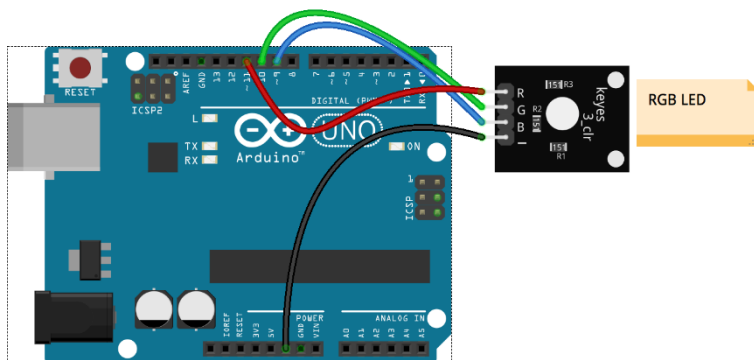


Krok 3: Zkompiluj kód

Nyní klikni na tlačítko sériového monitoru v pravém horním rohu na Arduino IDE. Objeví se okénko sériového monitoru. Teď otoč potenciometr a uvidíš na monitoru napětí pinů A0 na Arduino. Toto napětí bude odpovídat změně provedené potenciometrem.



RGB LED module	Arduino
R	11
G	10
B	9
-	GND



```
// RGB LED  
// Email:laskarduino@gmail.com  
// Web:laskarduino.cz  
/*\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/*/  
  
const int redPin    = 11;      // číslo pinu červené barvy  
const int greenPin  = 10;     // číslo pinu zelené barvy  
const int bluePin   = 9;       // číslo pinu modré barvy  
  
void setup() {  
    pinMode(redPin, OUTPUT);    // nastavení pinu redPin jako výstupu  
    pinMode(greenPin, OUTPUT); // nastavení pinu greenPin jako výstupu  
    pinMode(bluePin, OUTPUT);   // nastavení pinu bluePin jako výstupu  
}  
  
void loop() {  
    // Základní barvy:  
    color(255, 0, 0); // zapnout červenou
```

```

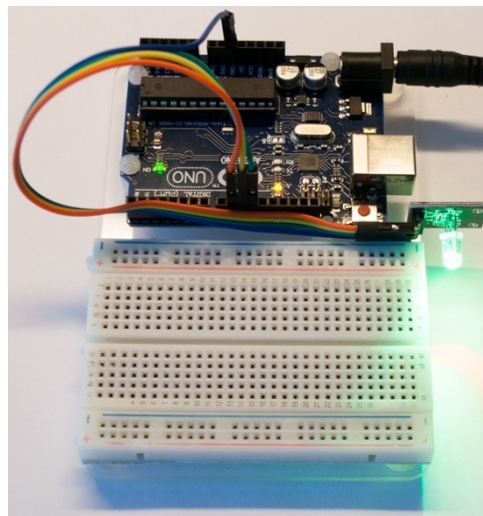
delay(1000); // počkat 1s
color(0,255, 0); // zapnout zelenou
delay(1000); // počkat 1s
color(0, 0, 255); // zapnout modrou
delay(1000); // počkat 1s
// Michaný barvy:
color(255,0,0); // červená
delay(1000);
color(237,109,0); // oranžová
delay(1000);
color(255,215,0); // žlutá
delay(1000);
color(0,255,0); // zelená
delay(1000);
color(0,0,255); // modrá
delay(1000);
color(0,46,90); // indigo
delay(1000);
color(128,0,128); // purpurová
delay(1000);
}
// generace barvy
void color (unsigned char red, unsigned char green, unsigned char blue)
{
    analogWrite(redPin, red);
    analogWrite(bluePin, blue);
    analogWrite(greenPin, green);
}

```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Nyní můžeš vidět, že RGB LED bliká červeně, zeleně, modře, oranžově, žlutě, fialově, indigo atd.



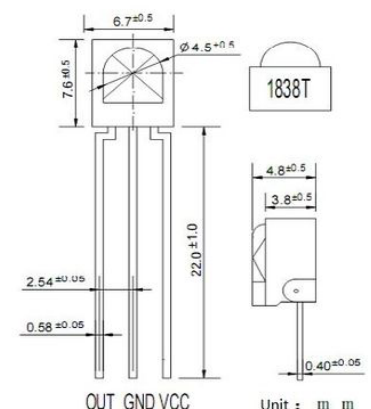
Lekce 13 Infračervený přijímač

Úvod

Hlavním prvkem tohoto infračerveného dálkového ovládání je přijímač HX1838. Tento infračervený přijímač umožňuje ve svém okolí detekovat signály z IR dálkových ovladačů a při jeho připojení k Arduino můžeme tyto signály převést na proměnné a dále využít. V našem Kitu je jen přijímač, spolu se kterým můžeš použít třeba dálkový ovladač od televize, kterému můžeme přiřadit libovolné funkce v programu.

Komponenty

- Arduino



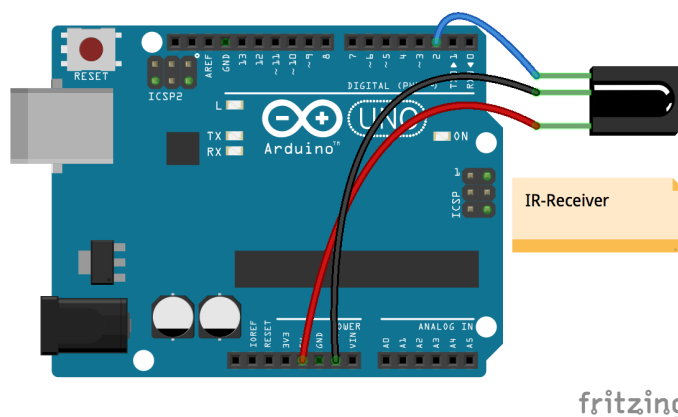
- USB kabel
- IR přijímač
- IR ovladač (není součástí kitu)
- Breadboard vodiče

Princip

Jako světelné zdroje se v dálkových ovladačích používají luminiscenční diody infra-LED, emitující paprsek záření o vlnové délce kolem 950nm. Jelikož při posílání informace způsobem log.1 = "infra-LED svítí", log.0 = "nesvítí" by bylo obtížné zajistit bezchybný přenos datového rámce z důvodu vysokého rušení (zdrojem takového infračerveného záření je i slunce či obyčejná žárovka). Využívá se tedy modulace.

Při modulaci logickou úrovní 1 se infra-LED budí obdélníkovým nosným signálem o dané střídě a frekvenci, typicky mezi 30-56kHz (38kHz v našem případě). Tento stav se nazývá značka.

Po dobu logické úrovně 0 není dioda aktivní, nastává mezera. Dobu trvání značky a mezery, stejně tak jako jejich význam, specifikuje použitý protokol. V klidu, tj. když neprobíhá přenos rámce, setrvává vysílač ve stavu log.0. Příjimač signálu je nastaven na použitý nosný kmitočet. Tím, že ostatní kmitočty odfiltruje, účinně zamezuje možnému rušení okolním světlem.



Postup experimentu

Krok 1: Vytvoř obvod

IR přijímač	Arduino
S	2
"_"	GND
+	5V

Krok 2: Napiš kód do Arduino IDE

Kód

```
// Infračervený přijímač
// Spustí seriálový monitor, vezmi ovladač.
// Stiskni klávesu ON/OFF a podívej se na kód v monitoru.
// My jsme používali ovladač od televize Philips, jeho ON/OFF kód je „0x4c“ a
„0x84c“.
// Ovladač přidává "8" každé sudé zmáčknutí.
// Vyměň náš ON/OFF kód za svůj a znova nahraj kód do Arduino.
// Když stiskneš klávesu ON / OFF, LED na Arduino se rozsvítí.
// Stiskni libovolnou jinou klávesu a LED zhasne.
// Email:laskarduino@gmail.com
// Web:laskarduino.cz
/*\/\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ /\ */
```

```
#include <IRremote.h>
```

```
const int IRPin = 2;           // číslo pinu tlačítka přijímače
const int ledPin = 13;        // číslo pinu zabudované LEDky

IRrecv irrecv(IRPin);         // vytvoříme objekt typu IRrecv
decode_results results;        // definujeme proměnnou

void setup() {
    pinMode(ledPin, OUTPUT);    // nastavení pinu LEDky jako výstupu
    Serial.begin(9600);         // spustit sériový monitor na 9600 bps
    irrecv.enableIRIn();        // zapnout infračervený přijímač
}

void loop() {
    if (irrecv.decode(&results)) { // pokud přijímač přijal data
```

```

Serial.print("IR Kod: "); // tisknout "irCode: "
Serial.print(results.value, HEX); // tisknout data jako hexadecimal
Serial.print(", bity: "); // tisknout " , bity: "
Serial.println(results.bits); // tisknout bity
irrecv.resume(); // přijata další data
}
delay(600); //počkat 600ms

// 2 Kódy tlačítka "ON/OFF"
if(results.value == 0x4c || results.value == 0x84c) {
    digitalWrite(ledPin,HIGH); // zapnout LEDku;
} else {
    digitalWrite(ledPin,LOW); // vypnout LEDku
}
}

```

```

// definujeme proměnné
int ledPin = 13; // definujeme pin LEDky

// Serial Monitor
// IR Kod: 4C, bity: 12
// IR Kod: 84C, bity: 12
// IR Kod: 4C, bity: 12
// IR Kod: 84C, bity: 12
// IR Kod: 4C, bity: 12

```

Krok 3: Zkompiluj kód

Krok 4: Nahraj sketch do Arduino

Spust' seriálový monitor, vezmi ovladač. Stiskni klávesu ON/OFF a podívej se na kód v monitoru. My jsme používali ovladač od televize Philips, jeho ON/OFF kód je „0x4c“ a „0x84c“. Ovladač přidává "8" každé sudé zmáčknutí. Vyměň náš ON/OFF kód za svůj a znova nahraj kód do Arduino. Když stiskneš klávesu ON / OFF, LED na Arduino se rozsvítí. Stiskni libovolnou jinou klávesu a LED zhasne.

