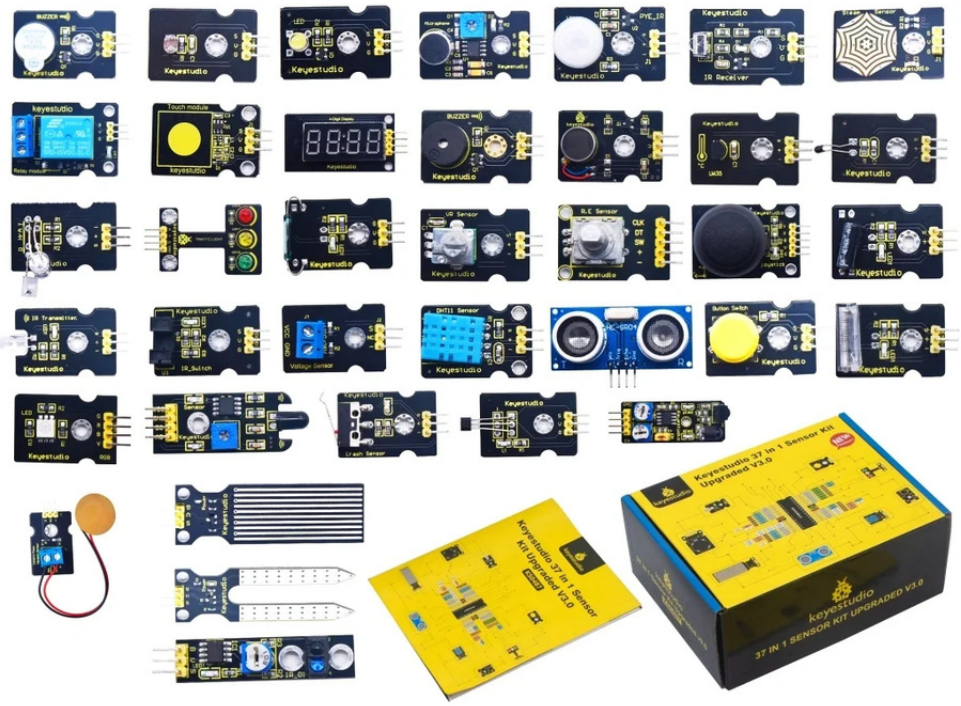
37 modulů a senzorů

Český návod pro Arduino

Keyestudio 37v1 V3.0 | KS0487 | HWKITCHEN KS005



Původní projekty, zapojení a obrázky Keyestudio. České vysvětlení, komentované programy, označené opravy a tipy HWKITCHEN.

Jak je návod uspořádán
Vydání 3.0 vychází ze struktury předlohy: úvod, součásti, instalace a 37 projektů. V každém projektu najdete popis, parametry, zapojení, program a očekávaný výsledek. Doplňky HWKITCHEN jsou zřetelně označeny.

9. listopadu 2026 • www.hwkitchen.cz

Obsah

1. Úvod 4
2. Seznam součástí 9
3. Instalace Arduino IDE a ovladačů 13
4. Projekty 15
01 Bílá LED 16
02 RGB LED 18
03 Semafor 20
04 Aktivní bzučák 22
05 Pasivní bzučák 24
06 Vibrační motorek 26
07 Tlačítko 28
08 Nárazový spínač 30
09 Sledování čáry 32
10 Infračervený detektor překážky 34
11 Optická závora 36
12 Hallův magnetický snímač 38
13 Snímač klepnutí 40
14 Digitální snímač náklonu 42
15 Kapacitní dotykový senzor 44
16 Snímač plamene a infračerveného záření 46
17 Jazýčkový spínač 48
18 PIR snímač pohybu 50

Obsah - pokračování

19 Analogový snímač teploty 52
20 Otočný potenciometr 55
21 Fotorezistor 57
22 Analogový snímač zvuku 59
23 Snímač hladiny vody 61
24 Snímač vlhkosti půdy 63
25 Snímač kapek a kondenzace 65
26 Keramický piezoelektrický snímač vibrací 67
27 Snímač napětí 69
28 Infračervený přijímač 71
29 Infračervený vysílač 73
30 Optický snímač pulzu 77
31 Joystick 79
32 Rotační enkodér 81
33 Jednokanálové relé 84
34 Lineární snímač teploty LM35 86
35 Teplota a vlhkost s DHT11 88
36 Ultrazvukový dálkoměr 90
37 Čtyřmístný LED displej 92
Přehled oprav a doplnění 94
Rychlé řešení problémů 95
Předloha, zpracování a zdroje 96

1. Úvod

Od prvního bliknutí k vlastnímu zařízení

Sada Keyestudio 37v1 V3.0 obsahuje 37 různých senzorů a dalších elektronických modulů. Některé zjišťují, co se děje v okolí: rozpoznají světlo, dotyk, pohyb nebo změnu teploty. Jiné dávají zařízení možnost reagovat: rozsvítí LED, vydají zvuk, rozvibrují motorek nebo zobrazí číslo. Postupně poznáte oba směry komunikace mezi programem a skutečným světem.

Představte si jednoduchého robota. Senzory jsou jeho smysly, program určuje jeho chování a mikrokontrolér vykonává pokyny programu. Výstupní prvky umožní robotovi odpovědět na okolí. Nejde o lidské přemýšlení: zařízení provádí konkrétní pravidla, která mu napíšete. Například „když se setmí, rozsviť světlo“. Stejný princip využijete v pokojové meteostanici, interaktivní hračce nebo školním pokusu.

Co je Arduino?

Arduino je otevřená platforma pro tvorbu elektronických zařízení. Patří k ní vývojové desky, programovací prostředí a knihovny. Slovo Arduino proto může podle souvislosti označovat celou platformu nebo konkrétní desku. V tomto návodu používáme Arduino UNO R3, případně odpovídající kompatibilní desku s mikrokontrolérem ATmega328P a 5V logikou.

Mikrokontrolér je malý počítač v jediném čipu: má procesor, paměť a obvody pro vstupy a výstupy. UNO k němu přidává konektory, napájení a USB rozhraní, takže jej můžete pohodlně zapojit a naprogramovat. Na běžném UNO R3 neběží Windows ani Linux. Po zapnutí se spustí nahraný program a deska jej vykonává, dokud má napájení.

Co se zde naučíte

Nejprve budete řídit světlo a zvuk. Potom přečtete tlačítka a senzory, zpracujete analogové hodnoty a vyzkoušíte přerušení, infračervenou komunikaci i displej. Všech 37 projektů zachovává pořadí předlohy Keyestudio. Není nutné znát vše předem: u každého projektu najdete zapojení, komentovaný program, vysvětlení a ověření výsledku.

Jak začít
Nejprve si přečtěte následující základy, připravte desku podle kapitoly 3 a vyzkoušejte projekt 1. Postupujte po jednom modulu. Nový pokus začínejte vždy jeho úplným programem a tabulkou zapojení; zkušenosti z předchozího modulu nemusí platit pro další.

1.1 Senzory, vstupy a výstupy

Vstup a výstup: vždy z pohledu Arduina

Senzor snímá fyzikální jev, například osvětlení nebo teplotu. Modul je malá deska, na které může být senzor doplněný rezistory a další elektronikou. Ne každý modul je senzor: LED, bzučák a displej slouží především jako výstupy. Pin je kovový elektrický vývod desky nebo modulu.

Vstup Arduina přijímá signál zvenčí. Výstup Arduina vysílá řídicí signál ven. Tentýž vodič tedy může spojovat výstup senzoru se vstupem Arduina. U LED modulu je směr opačný: výstup Arduina řídí vstup modulu. Pojmy „vstup“ a „senzor“ nejsou zaměnitelné.

Digitální signál: dva rozlišované stavy

Funkce digitalRead() vrací LOW nebo HIGH. U zde používané 5V desky odpovídá LOW nízkému napětí a HIGH dostatečně vysokému napětí. Nejde o měření přesného počtu voltů; rozlišují se povolená napěťová pásma. LOW ani HIGH samy neříkají, zda je senzor aktivní. Tlačítko může být při stisku LOW a dotykový snímač při dotyku HIGH. Rozhoduje konkrétní zapojení.

Analogový signál: velikost napětí

Analogový senzor převádí sledovaný jev na proměnné napětí. Analogově-digitální převodník (ADC) UNO R3 jej funkcí analogRead() převede na číslo od 0 do 1023, tedy na jednu z 1024 úrovní. Při výchozí referenci přibližně 5 V odpovídá jeden krok asi 4,9 mV. Rozlišení není totéž co přesnost: výsledek ovlivňuje i napájení, šum a tolerance součástek.

Číslo 600 z ADC ještě neznamená 600 °C, 600 luxů ani 60 % vlhkosti. Pro skutečnou jednotku potřebujete vztah mezi signálem a veličinou, někdy také kalibraci podle známých hodnot. Pro první pokus často stačí porovnat stav před změnou a po ní.

PWM: jak z digitálního výstupu řídit jas

analogWrite() na UNO R3 obvykle nevytváří plynulé stejnosměrné napětí. Na pinech označených vlnovkou ~ vytváří PWM: rychlé střídání LOW a HIGH s nastavitelným podílem doby HIGH. Hodnota 0 znamená trvale LOW, 255 trvale HIGH a 128 přibližně polovinu doby HIGH. Oko pak vnímá jiný jas LED. U modulu aktivního v LOW se účinek obrací.

Dva různé druhy digitálního připojení
Některé digitální moduly předávají jediný stav 0/1. Jiné posílají posloupnost bitů podle komunikačního protokolu. Proto teplotu z DHT11 nezískáte jedním digitalRead(): průběh komunikace zpracovává knihovna. Totéž platí pro dekódování IR ovladače.

1.2 Orientace na UNO R3

Napájecí a signálové vývody

GND je společná zem, vůči které měříme napětí. Vývod 5V používáme v těchto pokusech pro napájení vhodného modulu. Signálové piny D0 až D13 mohou sloužit jako digitální vstupy nebo výstupy. Jejich směr určuje pinMode(). Analogové vstupy jsou označeny A0 až A5. D3 a A3 jsou dva různé vývody.

PWM podporují piny D3, D5, D6, D9, D10 a D11. D0/RX a D1/TX jsou na UNO R3 propojené také se sériovým rozhraním k počítači; v těchto pokusech na ně moduly nepřipojujte. Tlačítko RESET spustí nahraný program od začátku. Program z paměti nezmizí po odpojení napájení, běžné proměnné v operační paměti se však při novém spuštění nastaví znovu.

Napětí, proud a odpor v několika větách

Napětí se udává ve voltech (V) a měří se mezi dvěma body. Elektrický proud udáváme v ampérech (A) nebo miliampérech (mA); 1000 mA = 1 A. Odpor se udává v ohmech (Ω), například 1 kΩ = 1000 Ω. Rezistor v sérii s obyčejnou LED omezuje proud. Hotový LED modul už potřebné součástky obsahuje; samostatnou LED nelze připojit stejně bez dalšího ověření.

Signálový výstup není univerzální zdroj napájení. Pro UNO R3 uvádí Arduino 20 mA na jeden I/O pin; omezený je i součet proudů více pinů. Holý motor ani cívku relé proto nenapájejte ze signálového pinu. Používejte modul s příslušným budicím obvodem. Současné připojení mnoha modulů vyžaduje samostatnou kontrolu spotřeby.

Označení pinů vždy čtěte v souvislostech

U běžného třípinového senzoru značí - nebo G zem, + nebo V napájení a S signál. Není to ale univerzální slovník pro všechny moduly: na RGB LED znamená G zelený kanál. DO/D0 na snímači znamená digitální výstup modulu, nikoli automaticky pin D0 Arduina. Řiďte se tabulkou v konkrétním projektu a potiskem svého kusu.

Pozor na dvě různá NC
U nezapojeného vývodu znamená NC „not connected“. U relé naopak NC znamená „normally closed“: kontakt spojený s COM, když není cívka buzena. NO je kontakt v klidu rozpojený. Zkratku NC tedy nelze obecně vykládat jako pokyn vývod nezapojovat. V projektech 26 a 27 zůstává jejich nevyužitý NC volný; v projektu 33 jde o označení přepínacího kontaktu.

1.3 Vybavení a bezpečné zapojení

Co si připravit

Potřebujete desku UNO R3 nebo odpovídající 5V desku s ATmega328P, počítač s Arduino IDE, datový USB kabel odpovídající konektoru desky a propojovací vodiče. Sada senzorů sama o sobě nenahrazuje řídicí desku; zkontrolujte obsah své zakoupené varianty. Originální UNO R3 používá USB typu B, kompatibilní desky mohou mít jiný konektor.

Pro moduly s kolíky a UNO se zásuvkami se obvykle hodí vodiče M/F (kolík/dutinka). Nepájivé kontaktní pole umožní rozdělit 5 V a GND do více větví. Pět otvorů ve stejné řadě bývá uvnitř propojeno, protilehlé poloviny přes středovou mezeru nikoli; napájecí lišty mohou být uprostřed přerušené. Konkrétní uspořádání nejprve ověřte.

U vybraných pokusů využijete magnet, proužek papíru, černou pásku, nádobku s vodou, zeminu, jeden 1,5V článek nebo IR ovladač. Projekt 29 pro komunikaci vyžaduje dvě samostatně naprogramované desky. Projekt 32 používá dva LED moduly; projekt 33 jeden. Použijte jednoduché LED moduly aktivní v HIGH; druhý takový modul může být potřeba doplnit.

Bezpečný postup v pěti krocích

1. Před každým přepojováním odpojte všechny zdroje: USB i případné externí napájení. Nestačí jen zavřít program v počítači.
2. Podle potisku identifikujte GND, napájení a signál. Barvy vodičů jsou pomůcka, nikoli elektrická vlastnost.
3. Zapojte zem, napájení a signály podle tabulky. Zkontrolujte zejména záměnu +/-, posun konektoru a spojení 5 V přímo s GND (zkrat).
4. Teprve po kontrole připojte USB a nahrajte správný program. Pokud se něco zahřívá nebo zapáchá, ihned napájení odpojte.
5. Měňte vždy jednu věc a výsledek si poznamenejte. Tak snadno zjistíte příčinu změny.

Voda, pohyb a různé napěťové úrovně

Elektroniku a konektory nenamáčejte. U vodních snímačů smáčejte pouze určenou měřicí plochu; ostatní části udržujte suché. Nepoužívejte horkou páru ani otevřený oheň. Relé a dělič napětí zkoušejte jen podle zdejších nízkonapěťových pokusů. Děti mají pracovat pod dohledem dospělého.

5 V není totéž co 3,3 V
Napájecí napětí modulu a napětí jeho signálu jsou dvě různé vlastnosti. Přenos na micro:bit, ESP32 nebo Raspberry Pi vyžaduje ověřit oba údaje, povolené vstupy a také program či knihovny. 5V signál může 3,3V vstup poškodit. Ani UNO R4 není z hlediska starších knihoven totožné s UNO R3.

1.4 Jak číst první program

Program v počítači a program v desce

Arduino IDE je aplikace v počítači, ve které píšete, překládáte a nahráváte program. Projektu se často říká sketch a zdrojový soubor má příponu .ino. Překlad ověří syntaxi a vytvoří strojový kód pro zvolenou desku. Nahrání jej přenese do paměti mikrokontroléru. Úspěšné nahrání ale ještě nepotvrzuje správné zapojení vodičů.

setup() a loop()

setup() se provede jednou po zapnutí nebo resetu. Nastavuje například směr pinů a rychlost sériové komunikace. Poté Arduino opakovaně vykonává loop(). Jakmile dojde na její konec, začne znovu. Složené závorky { } vymezují blok příkazů. Běžný příkaz končí středníkem; komentář začínající // platí do konce řádku.

Proměnné, podmínky a opakování

Proměnná má jméno, typ a hodnotu, kterou si program pamatuje. int na UNO R3 uchovává celé číslo od -32768 do 32767. Pro delší časové intervaly se běžně používá unsigned long. Podmínka if rozhodne, zda se určitý blok vykoná; else určuje druhou možnost. Smyčka for opakuje blok například desetkrát. Jedno = přiřazuje hodnotu, dvojité == hodnoty porovnává.

Nejpoužívanější funkce

pinMode(pin, OUTPUT) nastaví výstup; pinMode(pin, INPUT) vstup. digitalWrite(pin, HIGH) nastaví vysokou výstupní úroveň, digitalRead(pin) přečte digitální vstup. analogRead(A0) změří analogový vstup. analogWrite(pin, hodnota) řídí PWM na podporovaném pinu. Číslo pinu v programu musí souhlasit s fyzickým zapojením.

delay(1000) pozdrží hlavní program na jednu sekundu. Během čekání se další příkazy loop() neprovádějí, proto lze přehlédnout krátký stisk nebo zvuk. millis() vrací počet milisekund od spuštění a umožňuje načasovat akci bez takového čekání. Přerušení je zvláštní krátká obsluha události; podrobněji je vysvětleno v projektech 13 a 32.

Sériový monitor a knihovny

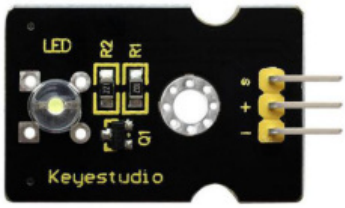
Serial.begin(9600) nastavuje sériovou komunikaci. Stejných 9600 baud vyberte v Sériovém monitoru, jinak mohou být znaky nečitelné. Serial.print() zapisuje text nebo číslo, Serial.println() navíc ukončí řádek. USB kabel zde přenáší data i napájí desku.

Knihovna je připravený soubor funkcí pro konkrétní úlohu. Řádek #include vloží její rozhraní; knihovna musí být nainstalovaná v odpovídající verzi. U projektů 28, 29, 35 a 37 proto nejprve postupujte podle oddílu Knihovny. Ostatní základní pokusy se obejdou bez těchto doplňků.

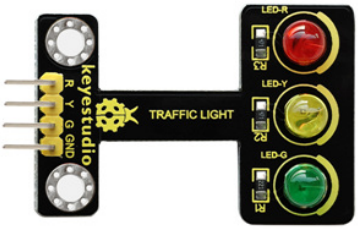
Pravidlo dobrého experimentu
Nejprve předpovězte, co program udělá. Potom jej spusťte a porovnejte předpověď s pozorováním. Teprve nakonec změňte jednu hodnotu, například prodlevu nebo práh. Zapište si původní hodnotu, novou hodnotu a výsledek. Tak se z kopírování programu stane porozumění.

2. Seznam součástí

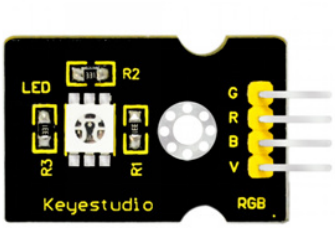
Přehled zachovává pořadí seznamu součástí v předloze. Číslo u každého modulu odkazuje na jeho projekt. Fotografie a zapojení v jednotlivých projektech pocházejí z dokumentace Keyestudio. Potisk a vzhled se mohou mezi výrobními revizemi mírně lišit.



Bílá LED Projekt 01



Semafor Projekt 03



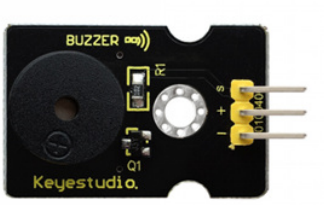
RGB LED Projekt 02



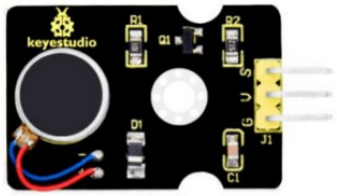
Čtyřmístný LED displej Projekt 37



Aktivní bzučák Projekt 04



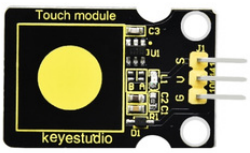
Pasivní bzučák Projekt 05



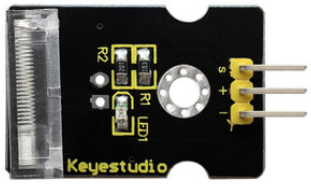
Vibrační motorek Projekt 06



Tlačítko Projekt 07



Kapacitní dotykový senzor Projekt 15



Snímač klepnutí Projekt 13

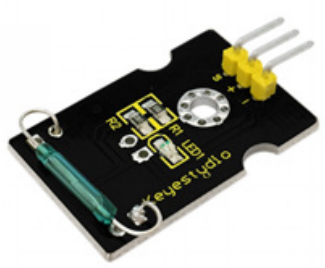
2. Seznam součástí - pokračování



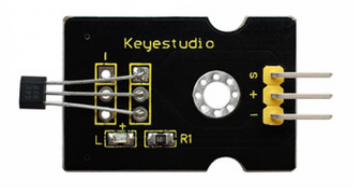
Digitální snímač náklonu Projekt 14



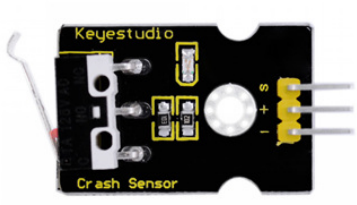
Optická závora Projekt 11



Jazýčkový spínač Projekt 17



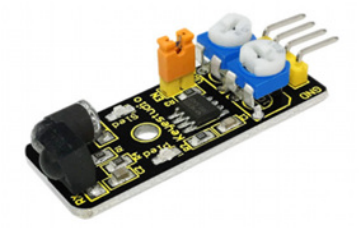
Hallův magnetický snímač Projekt 12



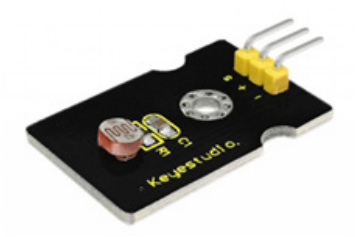
Nárazový spínač Projekt 08



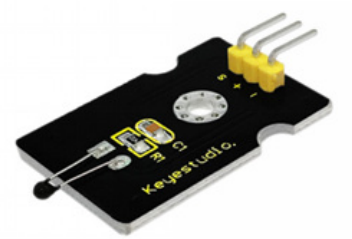
Sledování čáry Projekt 09



Infračervený detektor překážky Projekt 10



Fotorezistor Projekt 21

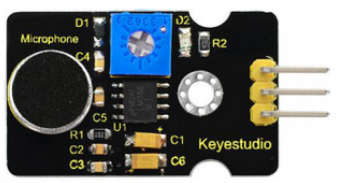


Analogový snímač teploty Projekt 19



Otočný potenciometr Projekt 20

2. Seznam součástí - pokračování



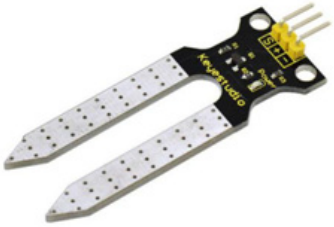
Analogový snímač zvuku Projekt 22



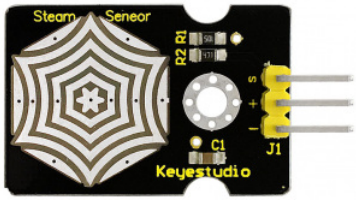
Snímač plamene a infračerveného záření Projekt 16



Snímač hladiny vody Projekt 23



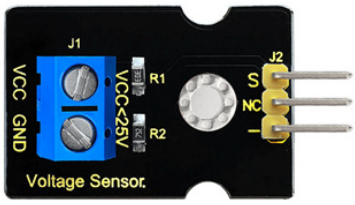
Snímač vlhkosti půdy Projekt 24



Snímač kapek a kondenzace Projekt 25



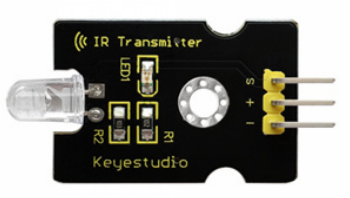
Keramický piezoelektrický snímač vibrací Projekt 26



Snímač napětí Projekt 27



Infračervený přijímač Projekt 28



Infračervený vysílač Projekt 29



Joystick Projekt 31

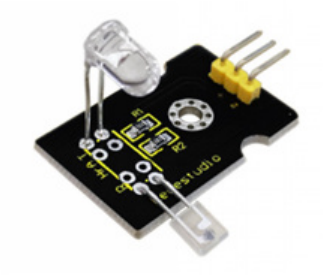
2. Seznam součástí - pokračování



Jednokanálové relé Projekt 33



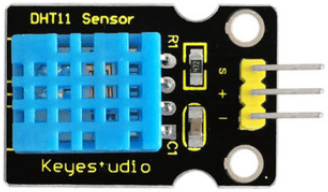
Rotační enkodér Projekt 32



Optický snímač pulzu Projekt 30



Lineární snímač teploty LM35 Projekt 34



Teplota a vlhkost s DHT11 Projekt 35



Ultrazvukový dálkoměr Projekt 36



PIR snímač pohybu Projekt 18

3. Instalace Arduino IDE a ovladačů

Předloha odkazuje na společného průvodce instalací IDE, ovladačů, nahráním programu a přidáním knihoven. Níže je doplněný český postup pro začátečníka.

Instalační průvodce z předlohy Keyestudio

Příprava desky a první nahrání

1. Nainstalujte Arduino IDE z oficiálního webu arduino.cc. Připojte UNO datovým USB kabelem. Nabíjecí kabel bez dat nestačí.

2. V nabídce desek zvolte Arduino Uno a doinstalujte balíček Arduino AVR Boards, pokud jej IDE nabídne. Vyberte port připojené desky.

3. Pokud port chybí, nejprve ověřte kabel a USB zásuvku. Ovladač USB převodníku instalujte podle skutečného čipu na konkrétní desce, z podkladů jejího výrobce.

4. V nabídce příkladů otevřete 01.Basics → Blink. Tlačítko Ověřit provede překlad, Nahrát navíc zapíše program do desky. Po nahrání má vestavěná LED L přibližně sekundu svítit a sekundu nesvítit. Až tento test funguje, odpojte napájení a začněte zapojovat moduly.

Programy a správné řádkování

Programy v PDF jsou skutečný text v neproporcionálním písmu. Nemají čísla řádků a žádný řádek kódu se při sazbě automaticky nezalamuje. Kopírujte vždy celý blok jednoho programu, od prvního komentáře po poslední závorku. V novém projektu nejprve označte a nahraďte všechen výchozí kód. Nevkládejte společně dvě funkce setup() a dvě loop().

Otevření přímo v Arduino IDE
Nejjednodušší cesta je doprovodný ZIP HWKITCHEN-KS005-programy-v3.zip. Obsahuje 40 souborů .ino ve správně pojmenovaných složkách. Rozbalte ZIP a otevřete vybraný soubor v Arduino IDE. Tím zůstane přesně zachováno odsazení a konce řádků. Stejný ZIP je vložen i jako příloha PDF; plnohodnotný PDF prohlížeč jej nabídne v panelu Přílohy.

Produkt KS005 - návody a programy ke stažení

Pokud kopírování v konkrétním PDF prohlížeči změní odsazení, použijte .ino soubor nebo Automatické formátování v Arduino IDE (obvykle Ctrl+T). Zkontrolujte hlavně, že komentář // nespolkl následující příkaz kvůli chybějícímu konci řádku.

Knihovny k původním programům

Knihovna musí odpovídat použitému rozhraní (názvům tříd a funkcí v programu). Nejnovější knihovna stejného nebo podobného názvu nemusí přeložit starší program. Zde základní ukázky záměrně zachovávají rozhraní předlohy.

Projekty 28 a 29 - IRremote

Používají IRremote.h, IRrecv, IRsend a decode_results. Jde o rozhraní řady 2.x; pro tato starší rozhraní zvolte IRremote 2.8.0. Verzi vybírejte ve Správci knihoven. Nepřepisujte pouze hlavičku na IRremote.hpp: nová řada 4.x má jiné rozhraní.

Projekt 35 - DHT11

Používá původní dht11.h, třídu dht11, metodu read() a konstanty DHTLIB_OK, DHTLIB_ERROR_CHECKSUM a DHTLIB_ERROR_TIMEOUT. Potřebujete původní knihovnu k sadě nebo knihovnu se shodným rozhraním. Adafruit DHT.h ani jiná náhodně vybraná knihovna DHT11 nejsou přímou náhradou.

Projekt 37 - TM1637

Používá TM1637.h a funkce init(), set(), point() a display(). Odpovídá mu původní knihovna k sadě a rozhraní knihovny Grove_4Digital_Display od Seeed Studio. Odlišná TM1637Display.h vyžaduje jiný program a v tomto základním návodu ji nepoužíváme.

Instalace ZIP knihovny

Je-li knihovna dodána v ZIP, v Arduino IDE použijte nabídku pro přidání ZIP knihovny. Při ruční instalaci patří adresář knihovny do složky libraries ve vašem sketchbooku; poté IDE znovu otevřete. Při hlášení více knihoven zkontrolujte, kterou cestu překladač skutečně použil.

Zdrojový projekt Keyestudio a složka CODE
Dokumentace KS0487 a rozcestník původních podkladů
Arduino-IRremote - dokumentace a verze
Seeed Studio - Grove_4Digital_Display

Rozsah ověření
U tohoto vydání byl porovnán kód s předlohou, zkontrolovány opravy, kopírovatelnost a struktura souborů. Neproběhlo fyzické ověření celé sady ani úplný překlad všech ukázek s původními knihovnami v Arduino IDE. Před zapojením vlastního zařízení nejprve ověřte jednotlivý pokus.

4. Projekty

Následuje všech 37 projektů ve stejném pořadí jako v předloze. Některé obsahují více ukázek: projekt 19 má dva programy, projekt 29 tři. Celkem je přiloženo 40 samostatných programů.

Postup u každého projektu

1. Poznejte modul podle fotografie a přečtěte jeho parametry.

2. Odpojte napájení a zapojte modul podle tabulky a původního obrázku. Označená oprava má přednost před chybným detailem originálního obrázku.

3. Otevřete příslušný soubor .ino nebo zkopírujte celý jeden blok programu. Doinstalujte požadovanou knihovnu.

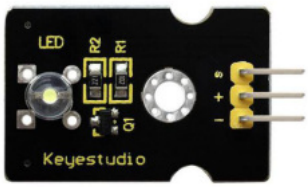
4. Program přeložte a nahrajte. Pro výpisy nastavte rychlost uvedenou v Serial.begin().

5. Porovnejte očekávaný výsledek. Teprve potom zkoušejte doplňující tip HWKITCHEN.

Jak číst obrázky
Fotografie výsledků jsou ilustrace z dokumentace výrobce. Nejde o záznam našeho fyzického testování. Pořadí pinů se může lišit mezi revizemi; jejich popisky mají přednost před barvou propojovacích vodičů.

Projekt 1: Bílá LED

Popis



LED modul je jednoduchý světelný výstup. Obsahuje diodu i součástky potřebné pro její řízení, takže jej lze připojit třemi vodiči k desce UNO. Stejný princip lze použít s jinou barvou LED modulu. V prvním pokusu se naučíte nastavit výstup a střídat jeho logické úrovně.

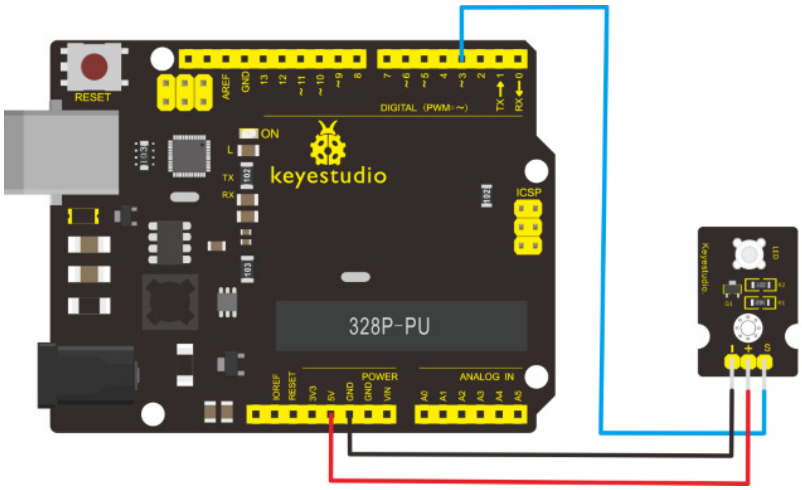
Parametry

- Rozhraní: digitální
- Konektor: rozteč 2,54 mm
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

pinMode() připraví D3 jako výstup. HIGH rozsvítí tento modul a LOW jej zhasne. Dvě delay(2000) tvoří celý cyklus dlouhý přibližně čtyři sekundy.

První ověření a častá chyba

Sledujte LED připojeného modulu, nikoli jen kontrolku L na UNO. Pokud nesvítí, zkontrolujte S na D3 a společnou zem; při kontrole odpojte napájení.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 1 - ukázkový program

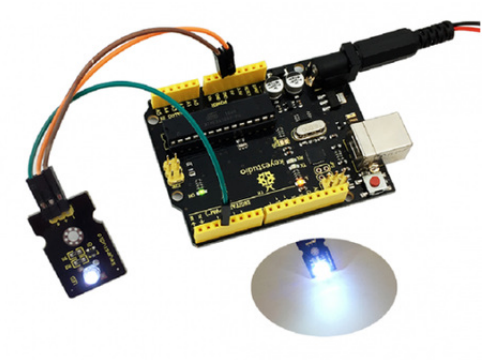
Projekt_01.ino

```
// Bílá LED - projekt 1
// D3 rozsvítí LED na 2 s a potom ji na 2 s zhasne.

int led = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(led, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  digitalWrite(led, HIGH);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(2000);
  digitalWrite(led, LOW);
  delay(2000);
}
```

Očekávaný výsledek

Po nahrání programu LED dvě sekundy svítí a dvě sekundy nesvítí. Tento cyklus se opakuje. Jde o stejné časování jako v programu předlohy.

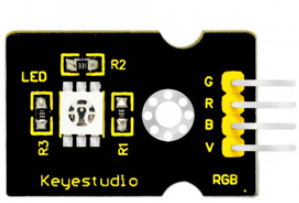


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Změňte pouze první delay(2000) na delay(200). Vysvětlete, proč se změní poměr doby svícení a zhasnutí. Samostatná LED bez rezistoru není náhradou za hotový modul.

Projekt 2: RGB LED

Popis



RGB modul spojuje červenou, zelenou a modrou LED. Řízením jednotlivých barevných složek lze vytvářet další odstíny. Předloha předvádí plynulou změnu hodnot PWM ve dvou navazujících smyčkách. Tuto podobu pokusu zachováváme. Modul s vývodem V má společné kladné napájení; menší hodnota PWM proto znamená delší dobu sepnutí příslušné barvy.

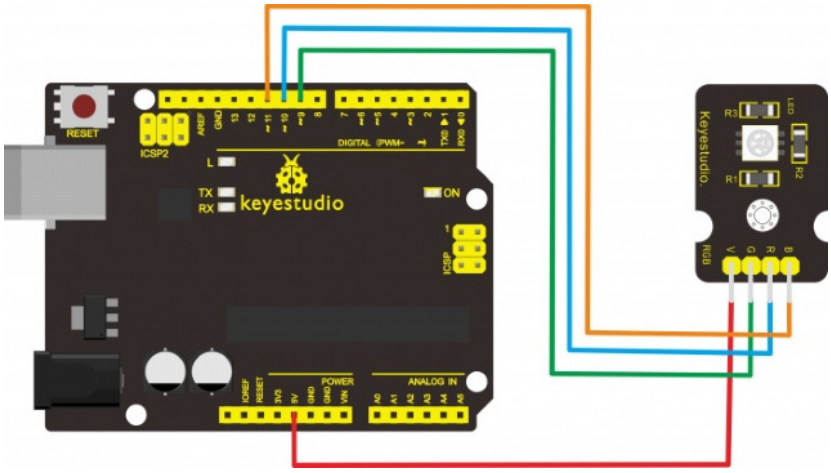
Parametry

- Napájení: 5 V
- Barvy: červená, zelená, modrá
- Řízení: digitální / PWM
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 4 propojovací vodiče.

Modul	Arduino / spoj
V	5 V
R	D10
G	D11
B	D9



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: pojmenování pinů je sjednoceno s původním obrázkem a textem (R = D10, G = D11, B = D9). Výraz 128 - val dával část času záporné PWM; nyní je omezen pomocí constrain na povolené hodnoty 0 až 255. Dvě původní prolínací smyčky i jejich časování zůstávají.

Jak pokus funguje

Tři LED se míchají do společné barvy. Smyčky for mění proměnnou val po jednom kroku. Protože jde o společnou anodu, PWM 0 kanál rozsvítí a 255 jej vypne. constrain() omezí výsledek do platného rozsahu.

První ověření a častá chyba

G na RGB modulu znamená Green, tedy zelenou barvu, nikoli GND. Piny musí být PWM. Ověřujte pomalé prolínání; pořadí vodičů R/G/B ovlivňuje výsledné odstíny.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 2 - ukázkový program

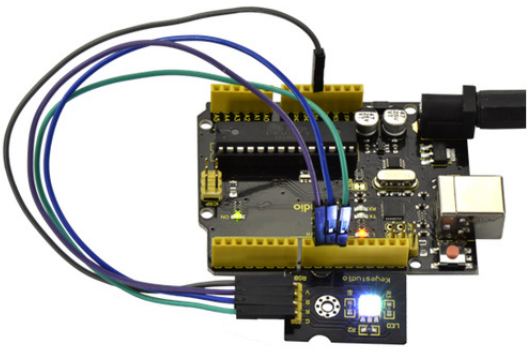
Projekt_02.ino

```
// RGB LED - projekt 2
// Dvě smyčky plynule mění PWM. V je společné kladné napájení.
// Pojmenování: R=D10, G=D11, B=D9; menší PWM více svítí.
// Oprava je popsána u projektu v českém návodu.

int redpin = 10;
int bluepin = 9;
int greenpin = 11;
int val;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(redpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(greenpin, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  for (val = 255; val > 0; val--) {
    // PWM používá hodnoty 0 až 255.
    analogWrite(11, val);
    analogWrite(10, 255 - val);
    analogWrite(9, constrain(128 - val, 0, 255));
    // Prodleva je v milisekundách: 1000 ms = 1 s.
    delay(1);
  }
  for (val = 0; val < 255; val++) {
    analogWrite(11, val);
    analogWrite(10, 255 - val);
    analogWrite(9, constrain(128 - val, 0, 255));
    delay(1);
  }
}
```

Očekávaný výsledek

Světlo plynule mění odstín. Hodnota val nejprve klesá od 255 a potom opět roste. Barvy nejsou přepínány na pevné sekundové intervaly.

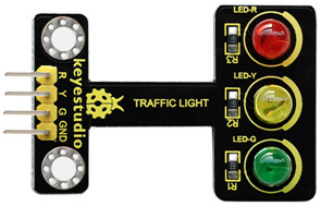


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Zvětšete delay(1) například na 10 ms a pozorujte pomalejší prolínání. Potom změňte pouze jeden barevný kanál a sledujte, které odstíny přestanou vznikat.

Projekt 3: Semafor

Popis



Modul obsahuje červenou, žlutou a zelenou LED v jednom celku. Pro každé světlo má samostatný řídicí vstup a společnou zem. Ukázka z předlohy nejprve rozsvítí zelenou, třikrát blikne žlutou a potom rozsvítí červenou. Je to demonstrace práce s několika výstupy a opakováním příkazů.

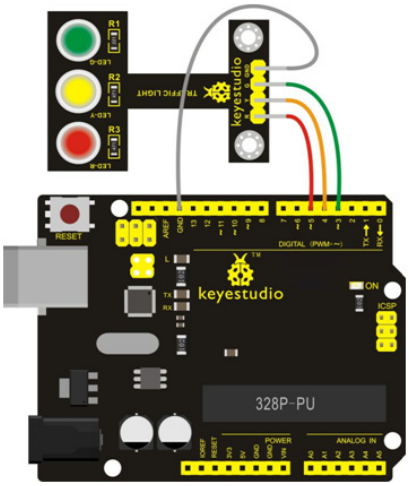
Parametry

- Napájecí / řídicí úrovně modulu: 3,3 až 5 V
- Rozhraní: digitální
- Konektor: rozteč 2,54 mm

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 4 propojovací vodiče.

Modul	Arduino / spoj
R	D5
Y	D4
G	D3
GND / -	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Každé světlo má vlastní digitální výstup. Smyčka třikrát zopakuje bliknutí žluté, zatímco ostatní příkazy řídí pořadí fází. Časování je postupné, po dobu delay() se další fáze neprovádějí.

První ověření a častá chyba

Ověřte nejprve barvy podle pinů R/Y/G. Společný GND není totéž co G pro zelenou. Sekvence je školní demonstrace, nikoli úplný program skutečné křižovatky.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 3 - ukázkový program

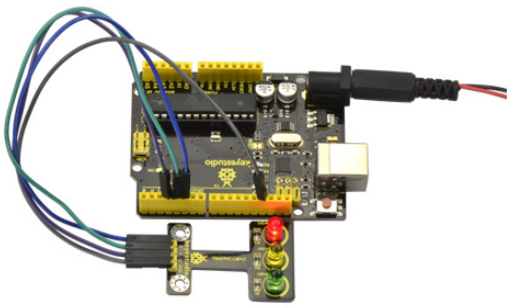
Projekt_03.ino

```
// Semafor - projekt 3
// Zelená 5 s, třikrát žluté bliknutí a červená 5 s.

int redled = 5;
int yellowled = 4;
int greenled = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(redled, OUTPUT);
  pinMode(yellowled, OUTPUT);
  pinMode(greenled, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  digitalWrite(greenled, HIGH);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(5000);
  digitalWrite(greenled, LOW);
  for (int i = 0; i < 3; i++) {
    delay(500);
    digitalWrite(yellowled, HIGH);
    delay(500);
    digitalWrite(yellowled, LOW);
  }
  delay(500);
  digitalWrite(redled, HIGH);
  delay(5000);
  digitalWrite(redled, LOW);
}
```

Očekávaný výsledek

Zelená svítí 5 sekund. Následují tři půlsekundová bliknutí žluté s přestávkami a poté 5 sekund červené. Zachována je i půlsekundová mezera před červenou.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Jako vlastní rozšíření přidejte po červené krátké současné svícení červené a žluté. Tím porovnáte původní demonstraci s běžnou sekvencí silničního semaforu.

Projekt 4: Aktivní bzučák

Popis



Aktivní bzučák obsahuje vlastní oscilátor. Přivedením řídicí úrovně jej lze zapínat a vypínat. Původní program velmi rychle střídá HIGH a LOW, vždy s prodlevou 1 ms. Program ponecháváme, aby odpovídal předloze; nejde o generování přesně naladěné melodie.

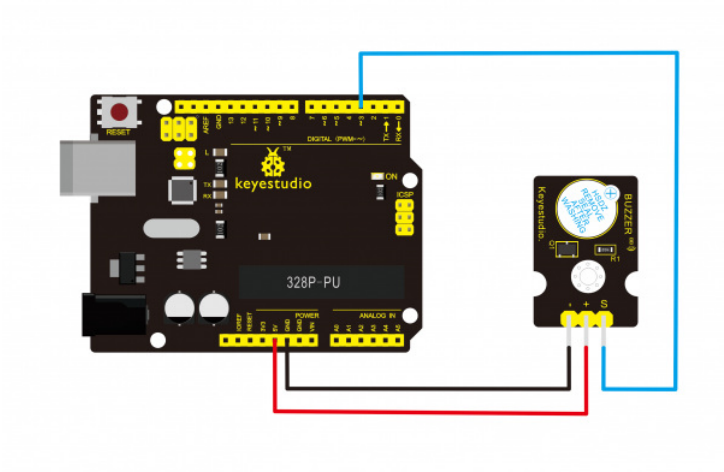
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: digitální
- Rozměry: 30 × 20 mm
- Hmotnost: 4 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Upřesnění HWKITCHEN: aktivní bzučák nelze popisovat stejně jako pasivní měnič. Změna rychlosti zapínání není obecně totéž jako přímé nastavování frekvence jeho vlastního tónu.

Jak pokus funguje

Oscilátor je obvod, který sám vytváří kmitání. Zde jej program pouze rychle zapíná a vypíná. Vlastnost aktivního bzučáku se lépe pozná při pomalém spínání uvedeném v tipu.

První ověření a častá chyba

Zkontrolujte, že máte aktivní modul z tohoto projektu. Záměna s pasivním mění zvuk i význam programu. Bzučák neposlouchejte z bezprostřední blízkosti ucha.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 4 - ukázkový program

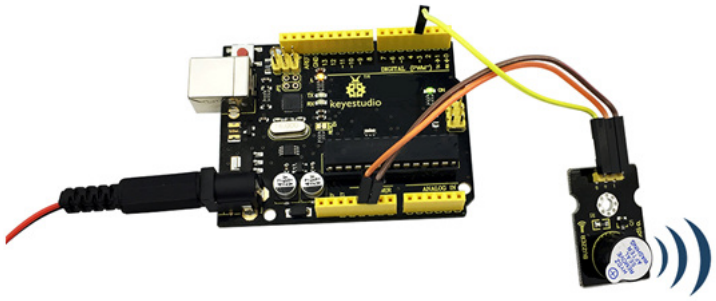
Projekt_04.ino

```
// Aktivní bzučák - projekt 4
// Aktivní bzučák má vlastní oscilátor; zde jej rychle spínáme.

int buzzPin = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(buzzPin, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  digitalWrite(buzzPin, HIGH);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(1);
  digitalWrite(buzzPin, LOW);
  delay(1);
}
```

Očekávaný výsledek

Bzučák vydává zvuk. Rychlé spínání ovlivňuje slyšený projev, ale základní tón aktivního bzučáku určuje jeho vnitřní obvod.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Zkuste obě prodlevy změnit na 500 ms. Uslyšíte zřetelné přerušované pípání. Pro samostatné tóny různých výšek použijte pasivní bzučák v následujícím projektu.

Projekt 5: Pasivní bzučák

Popis



Pasivní bzučák potřebuje střídavé buzení z programu. Výška tónu souvisí s periodou pulzů: kratší perioda znamená vyšší frekvenci. Předloha vytváří dvě série pulzů přímými zápisy na pin. Tento postup zachováváme a opravujeme chybějící prodlevu v druhé sérii.

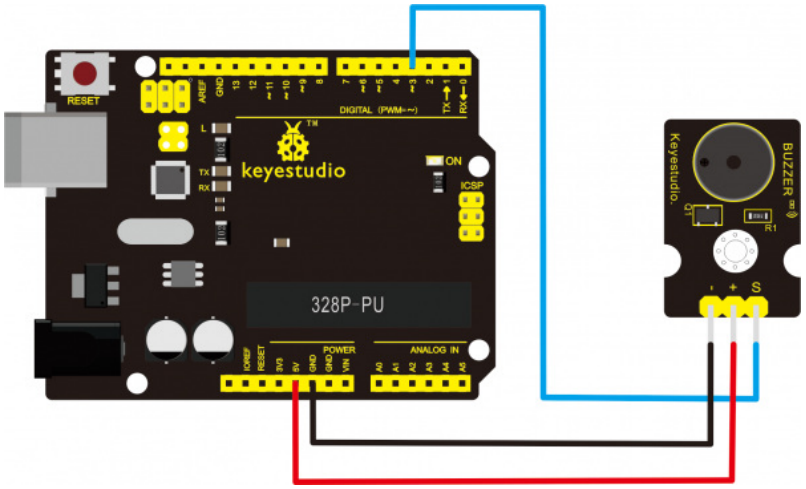
Parametry

- Napájení modulu: 3,3 až 5 V
- Rozhraní: digitální
- Rozměry: 30 × 20 mm
- Hmotnost: 4 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: ve druhé smyčce následoval LOW bezprostředně po HIGH. Doplněné delay(2) vytváří skutečnou kladnou půlperiodu. Zachovány jsou dvě série a jejich původní počty opakování.

Jak pokus funguje

Jeden úplný kmit tvoří HIGH i LOW. Frekvence je 1 / perioda: dvě milisekundy dávají přibližně 500 kmitů za sekundu. Počet opakování smyčky určuje délku zvukové části.

První ověření a častá chyba

Rozlišujte výšku tónu a dobu jeho trvání. Změna počtu opakování tón prodlouží, změna obou prodlev změní jeho výšku.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 5 - ukázkový program

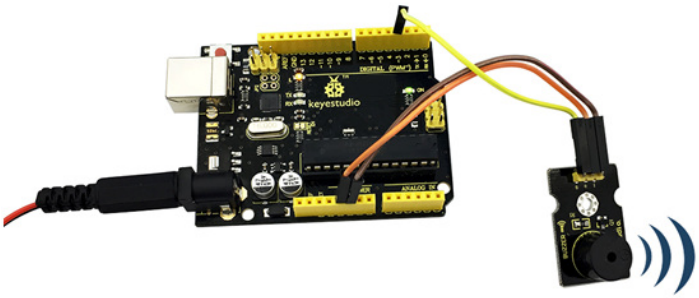
Projekt_05.ino

```
// Pasivní bzučák - projekt 5
// První série má periodu 2 ms, druhá po opravě 4 ms.
// Oprava je popsána u projektu v českém návodu.

int buzzer = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(buzzer, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  unsigned char i, j;
  while (1) {
    for (i = 0; i < 80; i++) {
      digitalWrite(buzzer, HIGH);
      // Prodleva je v milisekundách: 1000 ms = 1 s.
      delay(1);
      digitalWrite(buzzer, LOW);
      delay(1);
    }
    for (i = 0; i < 100; i++) {
      digitalWrite(buzzer, HIGH);
      delay(2);
      digitalWrite(buzzer, LOW);
      delay(2);
    }
  }
}
```

Očekávaný výsledek

Opakují se dvě zvukové části. První používá 1 ms HIGH a 1 ms LOW, přibližně 500 Hz. Druhá po opravě používá 2 ms HIGH a 2 ms LOW, přibližně 250 Hz. Skutečnou periodu mírně prodlouží vykonávání příkazů.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Zkuste změnit počet opakování 80 a 100. Tím upravíte délku zvuku, nikoli samotnou výšku tónu. Jako další pokus lze stejnou úlohu porovnat s funkcemi tone() a noTone().

Projekt 6: Vibrační motorek

Popis

Vibrační modul mění elektrické řízení na mechanické chvění. Hotová deska obsahuje potřebný budicí obvod; signálový pin Arduina není připojen přímo k holému motoru. Digitálním signálem lze vibrace zapnout a vypnout, PWM mění jejich intenzitu. Montážní otvor umožňuje připevnění do jednoduché hmatové pomůcky.

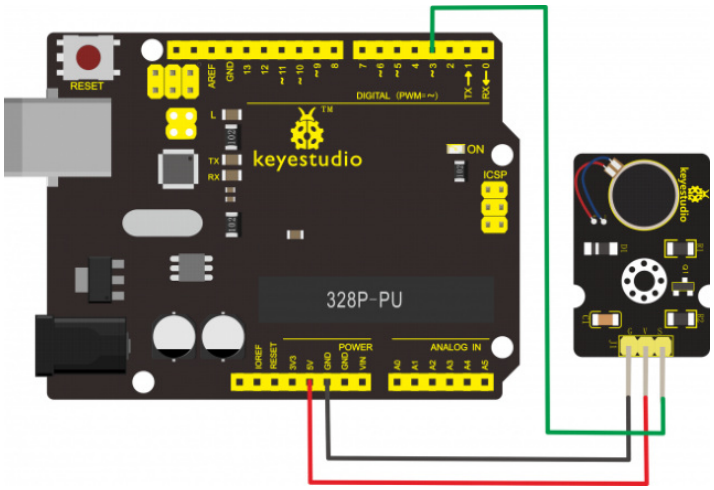
Parametry

- Napájení: 5 V DC
- Pracovní proud dle předlohy: 35 mA
- Maximální příkon dle předlohy: 0,5 W
- Teplota okolí: -25 až +65 °C
- Konektor: 3 piny, rozteč 2,54 mm
- Montážní otvor: 3 mm
- Rozměry: 34 × 19 × 6 mm; hmotnost 3,2 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

PWM se nemění na přesné nastavení síly v procentech. Program střídá dvě střídy a vypnutí; mechanická odezva motorku závisí také na upevnění a napájení.

První ověření a častá chyba

Položte modul tak, aby se vodivou částí nemohl dotknout ostatních spojů. Slabé PWM nemusí motor rozběhnout. Nikdy nenahrazujte budicí modul holým motorkem připojeným k D3.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 6 - ukázkový program

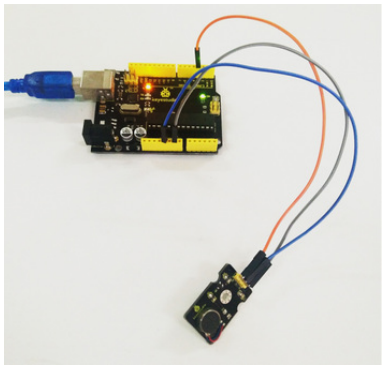
Projekt_06.ino

```
// Vibrační motorek - projekt 6
// PWM 200 a 100 střídá silnější a slabší vibrace motorku.

#define ZD 3
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ZD, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // PWM používá hodnoty 0 až 255.
  analogWrite(ZD, 200);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(5000);
  analogWrite(ZD, 0);
  delay(1000);
  analogWrite(ZD, 100);
  delay(5000);
  analogWrite(ZD, 0);
  delay(1000);
}
```

Očekávaný výsledek

Motorek 5 sekund vibruje s PWM 200, 1 sekundu stojí, 5 sekund vibruje s PWM 100 a opět 1 sekundu stojí. Celá sekvence se opakuje.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Připevněte modul na lehký karton a porovnejte vjem při PWM 100 a 200. Pokud se motorek při malé hodnotě nerozběhne, nemusí jít o chybu programu; motor potřebuje určitou rozběhovou energii.

Projekt 7: Tlačítko

Popis



Tlačítkový modul převádí stisk na digitální signál. Velké tlačítko, popsané vývody a montážní otvor usnadňují první pokusy i zapojení do senzorového shieldu. Program ovládá vestavěnou LED desky UNO podle stavu tlačítka.

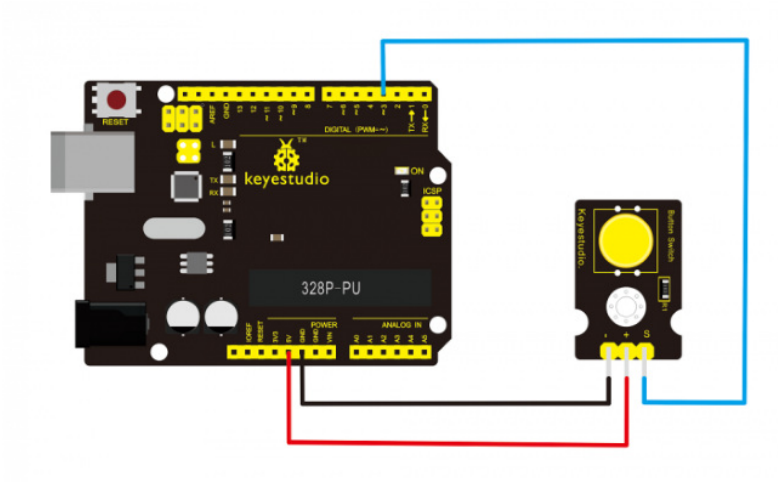
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: digitální
- Rozměry: 30 × 20 mm
- Hmotnost: 4 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Vstup je u tohoto modulu v klidu HIGH a při stisku LOW. Podmínka proto zhasíná LED při HIGH a rozsvěcí ji v opačné větvi. Elektronika modulu zajišťuje definovaný klidový stav.

První ověření a častá chyba

Pozorujte zvlášť stisk a uvolnění. Pro pouhé svícení nejsou zákmity obvykle problém; při počítání stisků by bylo nutné ošetřit krátké opakované změny kontaktu.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 7 - ukázkový program

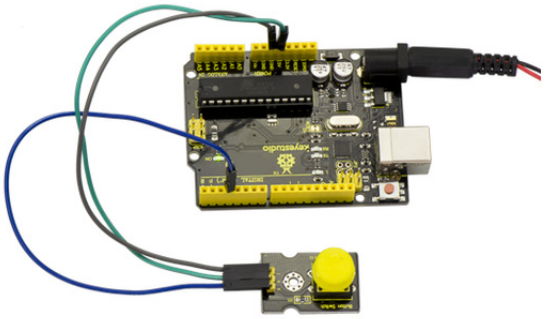
Projekt_07.ino

```
// Tlačítko - projekt 7
// Stisk tlačítka je LOW. Proto se LED rozsvěcí v else.

int ledPin = 13;
int inputPin = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ledPin, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(inputPin, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  int val = digitalRead(inputPin);
  if (val == HIGH) {
    digitalWrite(ledPin, LOW);
  }
  else {
    digitalWrite(ledPin, HIGH);
  }
}
```

Očekávaný výsledek

Při stisku je vstup LOW a vestavěná LED na D13 svítí. Při uvolnění je vstup HIGH a LED zhasne.

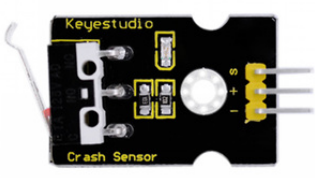


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Prozkoumejte rozdíl mezi „stisknuto = LOW“ a běžnou představou „zapnuto = 1“. Při rozšíření na počítadlo stisků bude potřeba ošetřit mechanické zákmity kontaktu.

Projekt 8: Nárazový spínač

Popis



Pružný mechanický spínač se hodí jako jednoduché čidlo dotyku s překážkou nebo koncový doraz. Při stlačení mění digitální výstup. Ukázka současně sleduje indikátor na modulu a vestavěnou LED Arduina. Otvor M3 slouží k upevnění, například na podvozek robota.

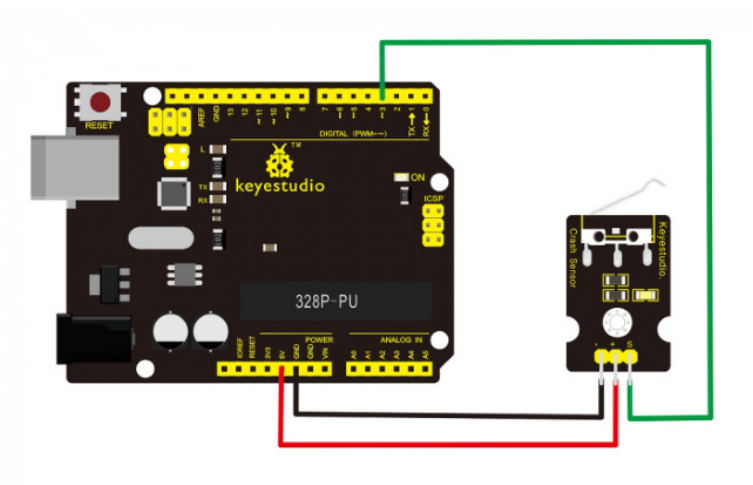
Parametry

- Aktivní stav při sepnutí: LOW
- Klidový stav: HIGH
- Napájení uvedené v předloze: 3 až 12 V; v tomto pokusu výhradně 5 V
- Rozměry: přibližně 31 × 21 mm
- Indikační LED a otvor M3

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Upřesnění HWKITCHEN: údaj 3 až 12 V u modulu není povolením přivést 12 V na vstup Arduina. Zde používáme napájení i logické úrovně 5 V.

Jak pokus funguje

Mechanický kontakt převádí náraz na sepnutí. Program čte okamžitý stav a nepamatuje si dřívější náraz. Slouží k indikaci dotyku, nikoli k měření jeho síly.

První ověření a častá chyba

Páčku stlačujte jemně a nepřetěžujte ji za mechanický doraz. Při krátkém klepnutí může indikace jen krátce probliknout.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 8 - ukázkový program

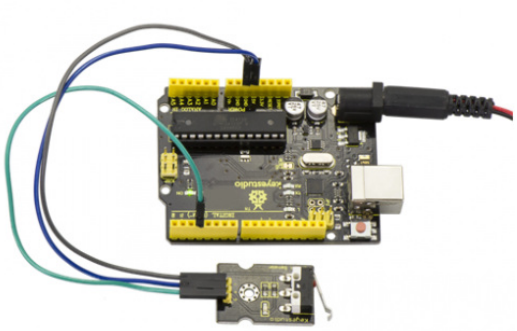
Projekt_08.ino

```
// Nárazový spínač - projekt 8
// Nárazový kontakt při sepnutí poskytuje LOW.

int Led = 13;
int Shock = 3;
int val;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(Led, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(Shock, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  val = digitalRead(Shock);
  if (val == HIGH) {
    digitalWrite(Led, LOW);
  }
  else {
    digitalWrite(Led, HIGH);
  }
}
```

Očekávaný výsledek

Po stlačení páčky se rozsvítí LED modulu a D13. Po uvolnění obě zhasnou.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Páčku přitlačujte prstem jen lehce. Jako rozšíření můžete přidat druhý spínač a rozlišovat levou a pravou překážku.

Projekt 9: Sledování čáry

Popis



Reflexní optický senzor rozlišuje povrchy podle množství odraženého infračerveného záření. Na výstupu poskytuje digitální hodnotu. Více podobných modulů může vytvořit snímací lištu robota sledujícího čáru. V této ukázce pouze čteme jeden snímač a sledujeme jeho reakci nad světlým a tmavým podkladem.

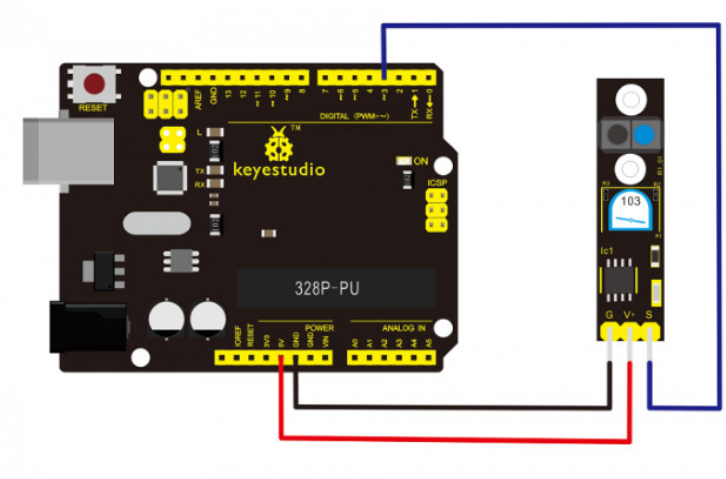
Parametry

- Napájení: 5 V
- Proud: méně než 10 mA
- Teplota okolí: 0 až +50 °C
- Výstup: digitální TTL
- Rozměry: 41,7 × 10,7 mm; hmotnost 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S / OUT	D3
V+	5 V
GND	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Senzor porovnává odraz s prahem a vrací 0/1. Černý povrch obvykle odráží méně než bílý, ale záleží i na materiálu, výšce a osvětlení. Program snímá jen dvakrát za sekundu.

První ověření a častá chyba

Držte stejnou malou výšku nad oběma plochami a zaznamenejte jejich stavy. Teprve podle tohoto testu určujte, která hodnota bude ve vašem programu znamenat čáru.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 9 - ukázkový program

Projekt_09.ino

```
// Sledování čáry - projekt 9
// Výstup senzoru čáry čteme jako 0 nebo 1, ne jako vzdálenost.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  Serial.println(digitalRead(3));
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(500);
}
```

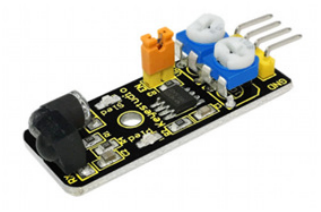
Očekávaný výsledek

V Sériovém monitoru při 9600 baud se každých 500 ms objeví 0 nebo 1. Přesouváním snímače mezi bílým papírem a černou páskou vyzkoušejte přechod mezi oběma stavy.

Tip HWKITCHEN - vyzkoušejte navíc
Udržujte stejnou malou vzdálenost od podkladu a doladťe trimr. Poznamenejte si, která hodnota odpovídá vašemu podkladu; lesk a okolní světlo mohou výsledek ovlivnit.

Projekt 10: Infračervený detektor překážky

Popis



Modul vysílá infračervené záření a vyhodnocuje jeho odraz. Trimry umožňují nastavit citlivost nebo práh detekce. Výstup oznamuje překročení prahu, neposílá vzdálenost v centimetrech. Předloha jej používá jako jednoduché oči robota, který má zpozorovat předmět před sebou.

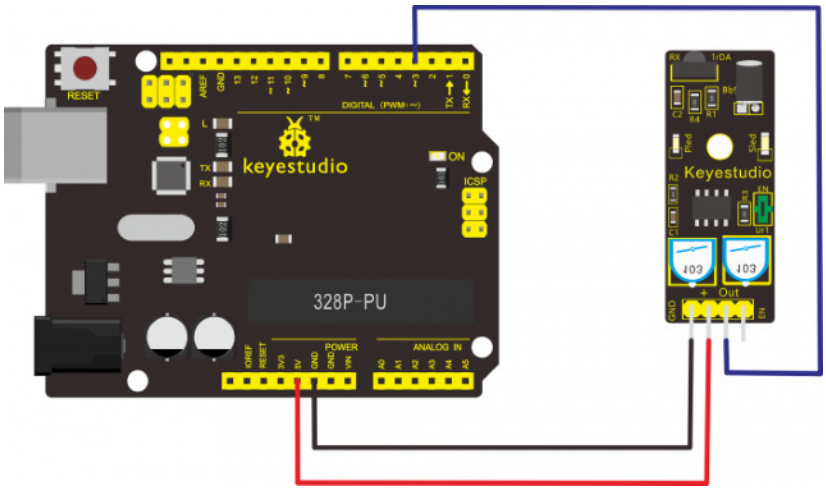
Parametry

- Napájení: 3,3 až 5 V
- Teplota okolí: -10 až +50 °C
- Uváděný dosah: přibližně 2 až 40 cm, závislý na povrchu
- Výstup: digitální TTL; některé revize mají i EN
- Uváděný úhel: 35°
- Rozměry: 41,7 × 16,7 mm; hmotnost 5 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
OUT	D3
V+	5 V
GND	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Upřesnění HWKITCHEN: původní kód i jeho polarita zůstávají. Nepřesný popis výsledku je nahrazen přesným vztahem mezi HIGH/LOW a D13. Použijte vývody OUT, V+ a GND; EN se řídí konkrétní revizí modulu.

Jak pokus funguje

Výstup vzniká porovnáním síly odrazu s nastaveným prahem. Tmavý nebo šikmý předmět může vracet méně světla než bílý. Digitální výsledek neobsahuje počet centimetrů.

První ověření a častá chyba

Citlivost nastavujte pomalu s kartičkou v požadované vzdálenosti. Pokud má modul EN, ověřte jeho funkci podle vlastní revize; nepřipojujte neznámý vývod jen podle barvy vodiče.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 10 - ukázkový program

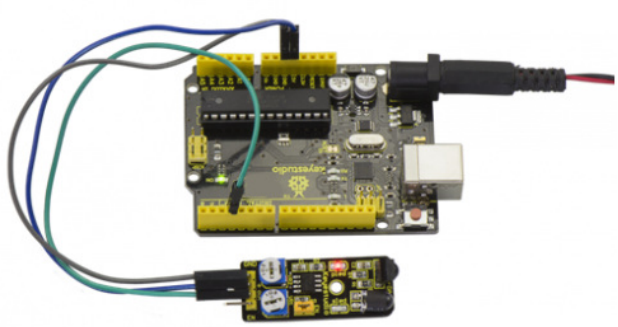
Projekt_10.ino

```
// Infračervený detektor překážky - projekt 10
// LED kopíruje signál senzoru. HIGH nemusí znamenat překážku.

const int sensorPin = 3;
const int ledPin = 13;
int sensorState = 0;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ledPin, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(sensorPin, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  sensorState = digitalRead(sensorPin);
  if (sensorState == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

Očekávaný výsledek

Vestavěná LED D13 kopíruje výstup: při HIGH svítí, při LOW zhasne. Přibližujte bílou kartičku a pozorujte také indikátor modulu. Stav LED proto nepopisujeme univerzálně jako „překážka = svítí“; záleží na polaritě konkrétního výstupu.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Porovnejte bílý papír, černý povrch a šikmo postavenou kartičku. Nastavený práh nebude představovat stejnou vzdálenost pro všechny předměty.

Projekt 11: Optická závora

Popis



V protilehlých ramenech štěrbiny je infračervený vysílač a přijímač. Vložený neprůhledný předmět přeruší světelnou cestu. Závora je vhodná pro jednoduchý doraz nebo počítání procházejících předmětů. Program zachovává původní přímé přiřazení signálu k LED.

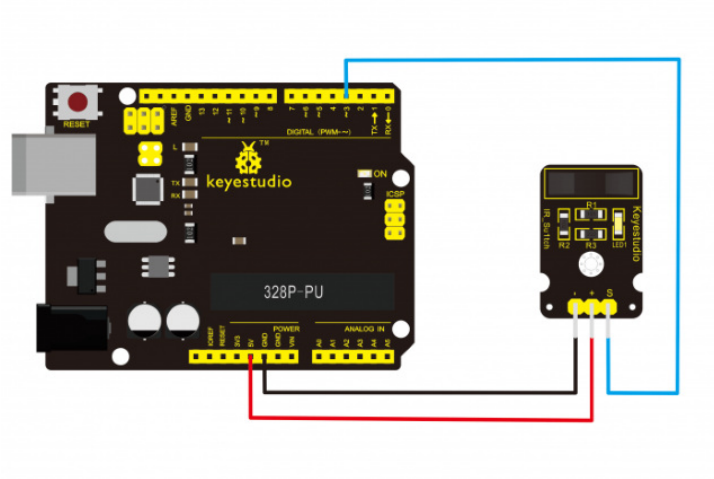
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: digitální
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava popisu HWKITCHEN: zavádějící komentář v originálním kódu je upřesněn. Logiku HIGH → rozsvícená D13 neměníme.

Jak pokus funguje

Paprsek přerušuje předmět ve štěrbině. Program pouze zobrazuje aktuální stav; pro počítání kusů by bylo nutné zachytit změnu stavu, ne přičítat v každém průchodu loop().

První ověření a častá chyba

Použijte neprůhledný proužek. Průhledný plast nemusí paprsek dostatečně přerušit. Při testu se nedotýkejte optických částí.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 11 - ukázkový program

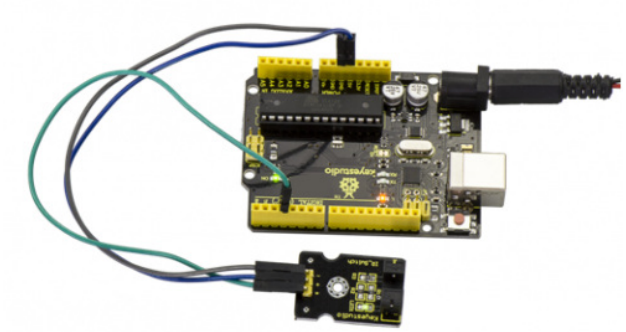
Projekt_11.ino

```
// Optická závora - projekt 11
// LED D13 kopíruje D3. Přerušením papírem se stav změní.

int Led = 13 ;
int buttonpin = 3;
int val ;
// setup() se provede jednou po zapnutí nebo resetu.
void setup () {
  // Řídicí pin nastavíme jako výstup.
  pinMode (Led, OUTPUT) ;
  // Senzorový pin bude vstupem.
  pinMode (buttonpin, INPUT) ;
}
// loop() Arduino stále znovu opakuje.
void loop () {
  val = digitalRead (buttonpin) ;
  if (val == HIGH) {
    digitalWrite (Led, HIGH);
  }
  else {
    digitalWrite (Led, LOW);
  }
}
```

Očekávaný výsledek

Při HIGH na vstupu D3 svítí D13, při LOW zhasne. Zasuňte do štěrbiny proužek papíru a vyjměte jej. U zobrazeného modulu v předloze při přerušení papírem indikace zhasíná.

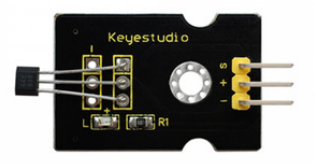


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Vystřihněte kartonový kotouč se štěrbinou a zkuste ji pomalu protáčet mezi rameny závory. Pro skutečné počítání otáček je potřeba počítat přechody, nikoli každý průchod funkcí loop().

Projekt 12: Hallův magnetický snímač

Popis



Hallův snímač reaguje na magnetické pole polovodičovou součástkou. Z modulu vychází digitální signál, který lze využít například pro detekci přítomnosti magnetu. Citlivost závisí na síle magnetu, jeho vzdálenosti a orientaci pólu.

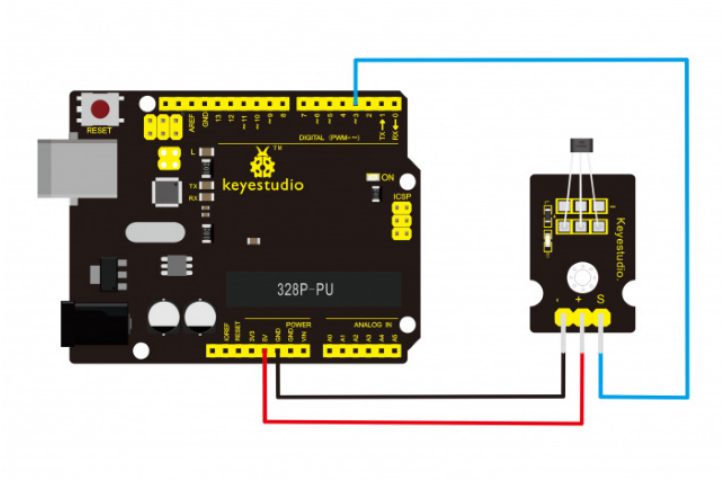
Parametry

- Výstup: digitální
- Dosah uvedený v předloze: až přibližně 3 cm, podle magnetu
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g
- Napájení v tomto zapojení: 5 V

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava textu HWKITCHEN: předloha zde zaměňuje Hallův snímač s jazýčkovým kontaktem. Jde o odlišné principy; jazýčkový spínač má vlastní kapitolu 17.

Jak pokus funguje

Hallův jev umožňuje snímat magnetické pole bez pohyblivého kontaktu. Digitální modul vyhodnocuje překročení prahu, neměří touto ukázkou velikost pole v teslech.

První ověření a častá chyba

Otestujte oba póly magnetu a několik vzdáleností. Označte si polohu, při které spíná. Silné magnety udržujte mimo citlivá zařízení a malé děti.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 12 - ukázkový program

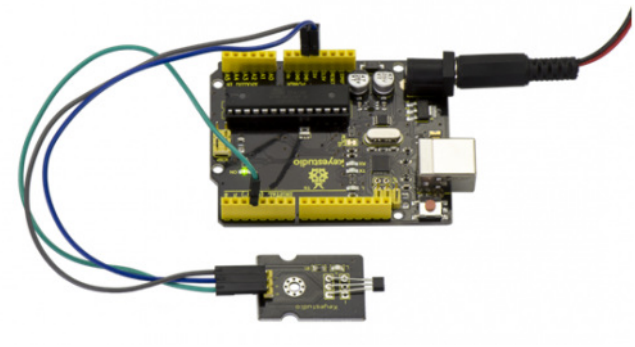
Projekt_12.ino

```
// Hallův magnetický snímač - projekt 12
// Hallův modul může při přiblížení magnetu přejít na LOW.

int ledPin = 13;
int inputPin = 3;
int val = 0;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ledPin, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(inputPin, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  val = digitalRead(inputPin);
  if (val == HIGH) {
    digitalWrite(ledPin, LOW);
  }
  else {
    digitalWrite(ledPin, HIGH);
  }
}
```

Očekávaný výsledek

V klidu D13 nesvítí. Při přiblížení vhodného pólu magnetu se může výstup přepnout na LOW a program D13 rozsvítí. Pokud změnu nevidíte, otočte magnet.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Najděte vzdálenost sepnutí a rozepnutí pro oba póly magnetu. Porovnejte výsledek s jazýčkovým spínačem v projektu 17.

Projekt 13: Snímač klepnutí

Popis



Otřesový kontakt při klepnutí vytvoří krátkou změnu digitálního signálu. Předloha používá přerušení na D3 a požaduje rozsvícení LED na dvě sekundy. Krátká událost se proto musí zachytit i mimo pravidelné čtení hlavní smyčky. Časování po opravě probíhá v loop(), zatímco obsluha přerušení pouze uloží příznak.

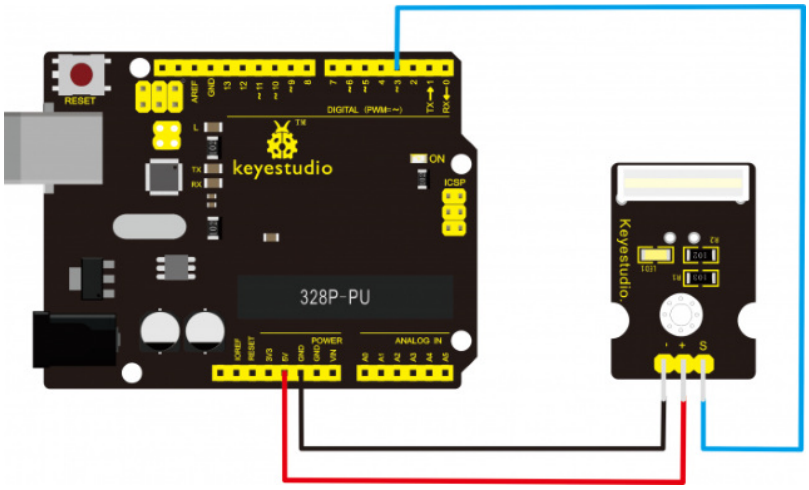
Parametry

- Napájení: 5 V
- Výstup: krátký digitální impuls
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: původní delay(2000) uvnitř přerušení spolehlivě nefunguje. Nahrazen je příznakem volatile a měřením času pomocí millis() mimo přerušení. Zapojení a požadovaný dvousekundový výsledek jsou zachovány.

Jak pokus funguje

Přerušení zachytí sestupnou hranu na D3. Obsluha nastaví příznak a rychle skončí. loop() příznak převezme a pomocí millis() udržuje LED dvě sekundy od poslední události.

První ověření a častá chyba

Snímač musí vytvořit novou hranu, trvalá úroveň sama další událost neznamená. Klepněte znovu během svícení a ověřte prodloužení doby. Čekání delay() do obsluhy přerušení nepatří.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 13 - ukázkový program

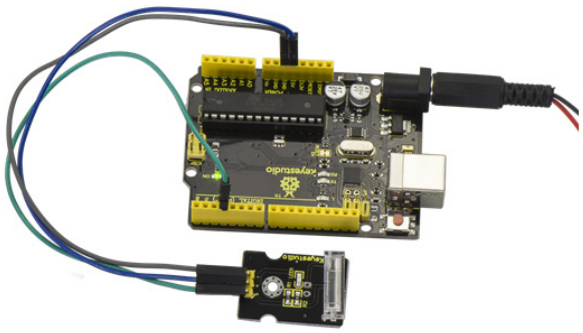
Projekt_13.ino

```
// Snímač klepnutí - projekt 13
// D3 odpovídá na UNO přerušení číslo 1.
// Obsluha přerušení jen uloží příznak; 2 s měří hlavní smyčka.
// Oprava je popsána u projektu v českém návodu.

const int interruptA = 1;
int LED = 13;
volatile bool klepnuti = false;
bool sviti = false;
unsigned long posledniKlep = 0;
void StateChanged();
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(LED, OUTPUT);
  attachInterrupt(interruptA, StateChanged, FALLING);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Krátce ochráníme sdílená data před přerušením.
  noInterrupts();
  bool noveKlepnuti = klepnuti;
  klepnuti = false;
  // Znovu povolíme přerušení.
  interrupts();
  if (noveKlepnuti) {
    posledniKlep = millis();
    sviti = true;
    digitalWrite(LED, HIGH);
  }
  if (sviti && millis() - posledniKlep >= 2000UL) {
    digitalWrite(LED, LOW);
    sviti = false;
  }
}
// Obsluha přerušení musí být krátká; nepoužívá delay().
void StateChanged() {
  klepnuti = true;
}
```

Očekávaný výsledek

Po jemném klepnutí D13 svítí dvě sekundy. Další zachycené klepnutí začne tuto dobu počítat znovu. Snímač neudeřujte velkou silou.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Jako rozšíření přidejte počítadlo klepnutí nebo krátké pípnutí. V obsluze přerušení pouze zaznamenejte událost; zvuk a výpis řešte v hlavní smyčce.

Projekt 14: Digitální snímač náklonu

Popis



Náklonový spínač mění stav kontaktu při změně polohy. Poskytuje jednoduchou informaci o překlopení, nikoli změřený úhel. Lze jej připojit přímo vodiči nebo přes senzorový shield. Program používá vestavěnou LED jako indikátor stavu.

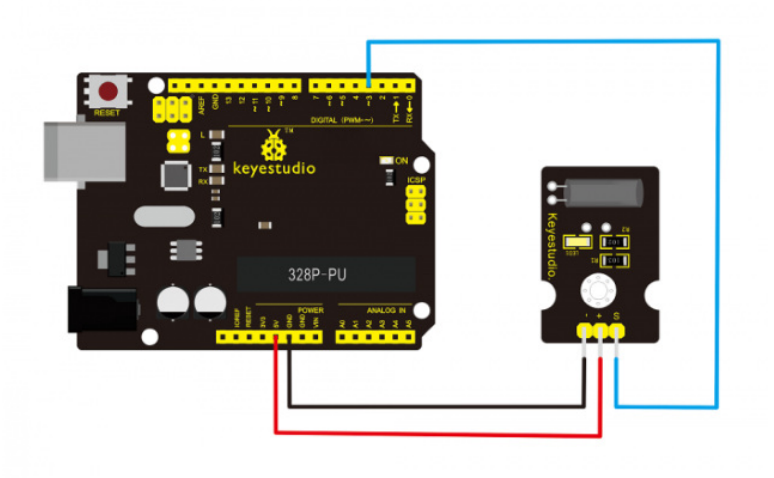
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: digitální
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Uvnitř se mechanicky změní kontakt. Stav se může poblíž překlápěcí polohy rychle měnit; nejde o akcelerometr ani přesný úhloměr.

První ověření a častá chyba

Nejprve podržte modul v obou krajních polohách. Krátké blikání při přechodu může být vlastnost mechanického kontaktu, nikoli chyba programu.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 14 - ukázkový program

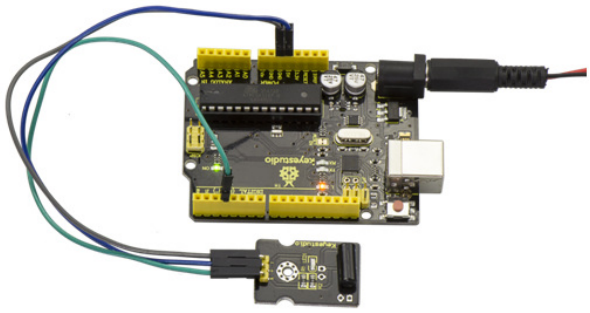
Projekt_14.ino

```
// Digitální snímač náklonu - projekt 14
// Čteme sepnutí kontaktu; program neměří úhel náklonu.

int ledPin = 13;
int switcher = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ledPin, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(switcher, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  if (digitalRead(switcher) == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

Očekávaný výsledek

Pomalu naklánějte modul. Při HIGH na D3 se rozsvítí D13; při LOW zhasne. Změna nastane při překročení mechanické polohy kontaktu.

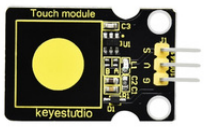


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Porovnejte pomalý náklon a rychlé zatřesení. Krátké blikání při změně polohy ukazuje, proč se v hotovém zařízení používá časové filtrování.

Projekt 15: Kapacitní dotykový senzor

Popis



Dotykový modul rozpoznává změnu kapacity v okolí snímací plochy. Mechanické stisknutí není nutné. Může reagovat i přes tenkou vrstvu papíru nebo látky, přičemž silnější překážka citlivost obvykle snižuje. Výstup je digitální a ukázka jím ovládá D13.

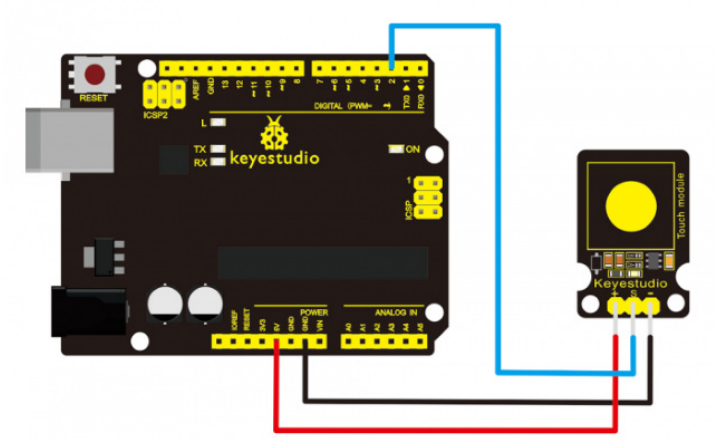
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: digitální
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D2
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Prst ovlivní elektrickou kapacitu snímací plochy. Elektronika tento rozdíl vyhodnotí a program převede digitální stav na světlo.

První ověření a častá chyba

Po zapnutí se plochy chvíli nedotýkejte. Otestujte nejprve přímý dotyk, potom tenký suchý papír. Dlouhé vodiče a vlhkost mohou chování změnit.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 15 - ukázkový program

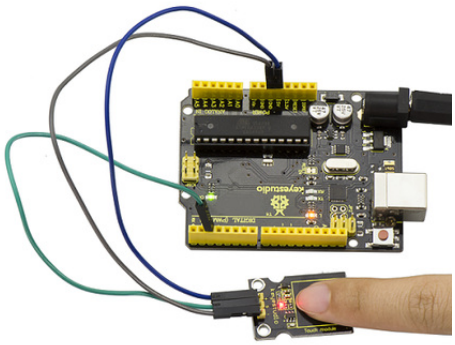
Projekt_15.ino

```
// Kapacitní dotykový senzor - projekt 15
// Dotykový signál HIGH na D2 rozsvítí vestavěnou LED D13.

int ledPin = 13;
int KEY = 2;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ledPin, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(KEY, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  if (digitalRead(KEY) == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

Očekávaný výsledek

Po dotyku snímací plochy se výstup přepne na HIGH a rozsvítí se D13 i indikátor modulu. Po oddálení prstu se indikace vrátí do klidu.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Vyzkoušejte jednu a několik vrstev papíru. Při připojování napájení nechte snímací plochu volnou, aby se modul mohl ustálit.

Projekt 16: Snímač plamene a infračerveného záření

Popis



Senzor reaguje na světlo v blízké infračervené oblasti. Plamen je jedním z možných zdrojů takového záření, reagovat ale mohou i jiné světelné zdroje. Nastavitelný práh umožňuje získat digitální výstup. Ukázka předlohy čte D0 na D2 a přímo řídí vestavěnou LED.

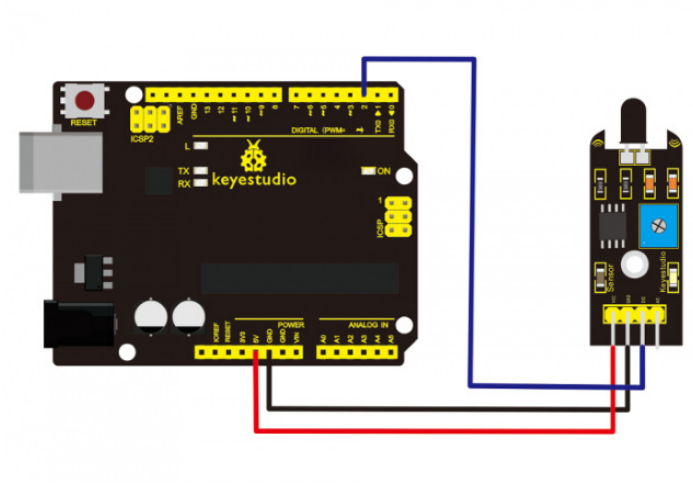
Parametry

- Napájení: 3,3 až 5 V
- Spektrální oblast dle předlohy: 760 až 1100 nm
- Teplota okolí dle předlohy: -25 až +85 °C
- Rozhraní: digitální
- Rozměry: 44 × 16,7 mm; hmotnost 4 g
- Detekce závisí na zdroji, vzdálenosti a nastavení prahu

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
D0 / DO	D2
VCC	5 V
GND	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Doplnění HWKITCHEN: původní fotografie ukazuje zapalovač. Pro dětské pokusy nepoužívejte otevřený oheň; jde o ilustraci výrobce. Modul není certifikovaný požární hlásič. Nejasný údaj o dosahu v původním textu není zárukou detekční vzdálenosti.

Jak pokus funguje

Komparátor na modulu porovná intenzitu záření s prahem. DO/D0 je výstup tohoto porovnání; v našem zapojení vede na D2 Arduina.

První ověření a častá chyba

Zkuste zdroj blízkého IR záření, například ovladač namířený zblízka; pulzy nemusí být vždy zachyceny. Běžná bílá LED svítilna nemusí stačit. Modul nikdy nepoužívejte jako jedinou požární ochranu.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 16 - ukázkový program

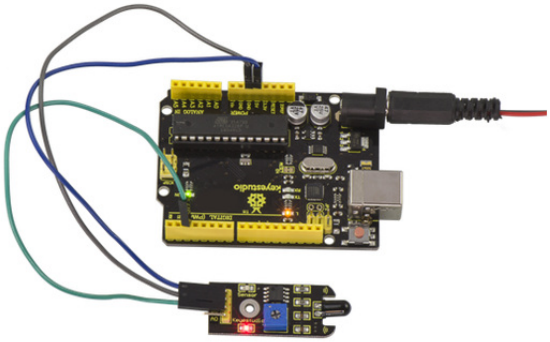
Projekt_16.ino

```
// Snímač plamene a infračerveného záření - projekt 16
// D0 snímače je připojen na D2. LED kopíruje jeho úroveň.

const int flamePin = 2;
const int ledPin = 13;
int State = 0;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ledPin, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(flamePin, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  State = digitalRead(flamePin);
  if (State == HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
}
```

Očekávaný výsledek

Při změně osvětlení může dojít k překlopení indikátoru modulu. Program rozsvítí D13 při HIGH a zhasne ji při LOW; konkrétní stav při detekci záleží na výstupní polaritě modulu.

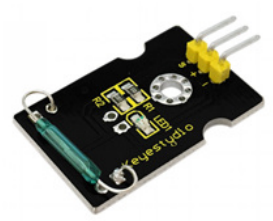


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Vyzkoušejte změnu okolního osvětlení a nastavení trimru. Některé IR ovladače mohou vyvolat reakci, ale jejich impulsy nejsou zaručeným testovacím zdrojem pro každé nastavení.

Projekt 17: Jazýčkový spínač

Popis



Ve skleněné trubičce jsou pružné feromagnetické kontakty. Magnetické pole může kontakty spojit; po oddálení magnetu se vracejí do původního stavu. Takové snímání se používá u dveří, jednoduchých počítadel nebo snímání otáček kola. Výsledný stav elektroniky modulu zjišťujeme digitálním vstupem.

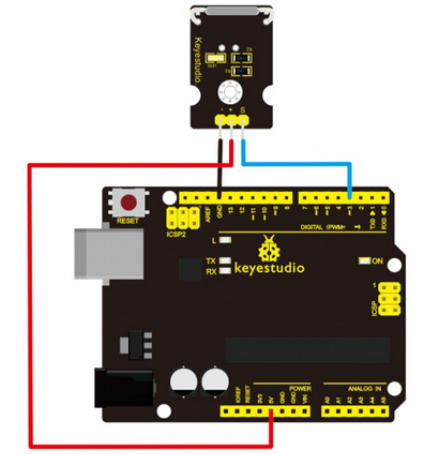
Parametry

- Napájení: 3,3 až 5 V
- Teplota okolí: -10 až +50 °C
- Uváděná detekční vzdálenost: do 10 mm, podle magnetu
- Vývody: S, +, -
- Rozměry: 30 × 20 mm; hmotnost 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Magnet ovládá fyzické kontakty v trubičce, zatímco Hallův snímač z projektu 12 pracuje polovodičově. Oba mohou do programu předávat jen dva stavy.

První ověření a častá chyba

Magnet přibližujte z různých stran. Skleněnou trubičku neohýbejte ani nestlačujte. Při počítání sepnutí počítejte také s odskoky kontaktů.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 17 - ukázkový program

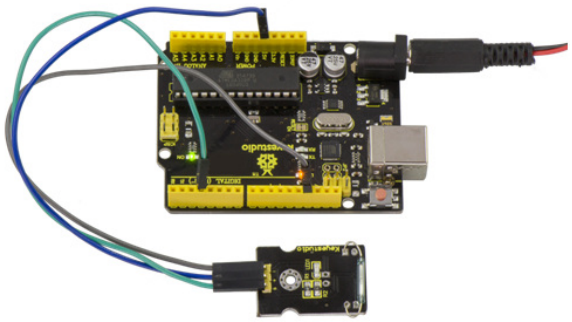
Projekt_17.ino

```
// Jazýčkový spínač - projekt 17
// Při HIGH je D13 zapnutá, při LOW zhasnutá.

int Led = 13;
int buttonpin = 3;
int val;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(Led, OUTPUT);
  // Senzorový pin bude vstupem.
  pinMode(buttonpin, INPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  val = digitalRead(buttonpin);
  if (val == HIGH) {
    digitalWrite(Led, HIGH);
  }
  else {
    digitalWrite(Led, LOW);
  }
}
```

Očekávaný výsledek

Předloha u zobrazeného modulu ukazuje v klidu rozsvícenou D13. Po přiblížení magnetu svítí indikátor modulu a D13 zhasne. Program kopíruje úroveň vstupu bez obrácení.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Vyrobte z kartonu model dveří: magnet dejte na pohyblivou část, modul na rám. Skleněnou trubičku neohýbejte a magnet na ni nepřikládejte nárazem.

Projekt 18: PIR snímač pohybu

Popis



PIR modul zaznamenává změny infračerveného záření způsobené například pohybem člověka nebo zvířete v jeho zorném poli. Vestavěná elektronika převádí změnu na časově omezený digitální signál. Snímač je určen k zachycení pohybu, nikoli ke spolehlivému potvrzení přítomnosti nehybné osoby.

Parametry

- Napájení: 3,3 až 5 V; maximum uvedené v předloze 6 V
- Proud uvedený v předloze: 15 µA
- Teplota okolí: -20 až +85 °C
- Výstup: přibližně 3 V při HIGH a 0 V při LOW
- Doba HIGH: přibližně 2,3 až 3 s
- Úhel: přibližně 100°; dosah do 7 m za vhodných podmínek
- Rozměry: 30 × 20 mm; hmotnost 4 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND

Oprava nebo upřesnění předlohy
Upřesnění HWKITCHEN: nejasný údaj „Pin limit current: 100 mA“ z předlohy nelze považovat za dovolený zatěžovací proud výstupu. Výstup zde připojujeme pouze na vstup Arduina.

Jak pokus funguje

PIR porovnává změny tepelného záření v částech zorného pole. Člověk pohybující se napříč ním vyvolá změnu snadněji než nehybná osoba.

První ověření a častá chyba

Po připojení vyčkejte na ustálení, které může trvat desítky sekund. Ověřujte pohyb ruky nebo těla, nikoli pouhou přítomnost. Text „No one!“ znamená jen neaktivní výstup.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 18 - ukázkový program

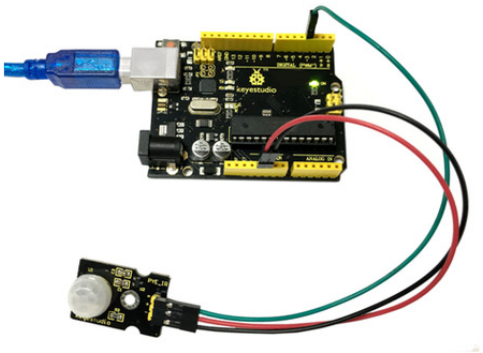
Projekt_18.ino

```
// PIR snímač pohybu - projekt 18
// PIR oznamuje pohyb logickou 1; LED kopíruje tento stav.

byte sensorPin = 3;
byte indicator = 13;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Senzorový pin bude vstupem.
  pinMode(sensorPin, INPUT);
  // Řídicí pin nastavíme jako výstup.
  pinMode(indicator, OUTPUT);
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  byte state = digitalRead(sensorPin);
  digitalWrite(indicator, state);
  if (state == 1)Serial.println("Somebody is in this area!");
  else if (state == 0)Serial.println("No one!");
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(500);
}
```

Očekávaný výsledek

V Sériovém monitoru při 9600 baud se při HIGH vypisuje „Somebody is in this area!“ a svítí D13. Při LOW se vypisuje „No one!“ a LED zhasne. Po zapnutí dejte snímači čas na ustálení.

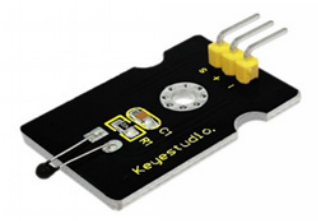


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Zkuste pohyb napříč zorným polem a pohyb přímo směrem k senzoru. Potom se zastavte: nehybný člověk může přestat vyvolávat signalizaci.

Projekt 19: Analogový snímač teploty

Popis



Modul používá termistor: jeho odpor se mění s teplotou. Napěťový dělič převádí změnu odporu na analogové napětí. Předloha nejprve ukazuje samotnou hodnotu ADC a poté druhý program pro výpočet teploty. Zachováváme obě ukázky ve stejném pořadí. Převod používá model NTC 10 kΩ při 25 °C, parametr B = 3950 K a druhý odpor děliče 4,7 kΩ. Tyto konstanty musí odpovídat skutečnému provedení modulu; samotný program jejich správnost neumí zjistit.

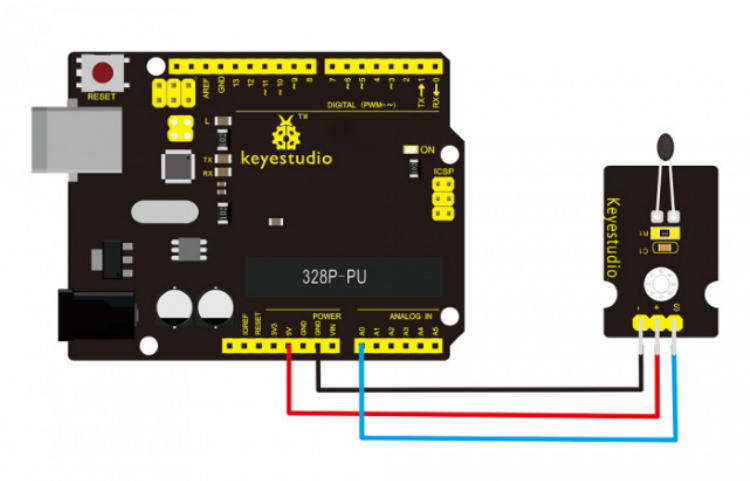
Parametry

- Napájení: 5 V
- Rozhraní: analogové
- Model v programu: R25 = 10 kΩ, B = 3950 K, odpor děliče 4,7 kΩ
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: druhý program odmítá krajní hodnoty ADC, při kterých by hrozilo dělení nulou nebo neplatný logaritmus. Široký rozsah -55 až +315 °C uvedený předlohou není ověřeným provozním rozsahem celého modulu; desku nevystavujte těmto teplotám.

Jak pokus funguje

První program ukáže pouze ADC. Druhý z napětí počítá odpor a z něj teplotu pomocí B rovnice. Konstanty 4,7 kΩ, 10 kΩ a B=3950 K jsou předpoklady převodu z předlohy.

První ověření a častá chyba

Nejprve porovnejte výsledek s pokojovým teploměrem. Pokud se výrazně liší, ověřte skutečné odpory a typ NTC; samotná ochrana krajních hodnot nesprávnou kalibraci neopraví.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 19 - ukázkový program 1/2

Projekt_19_ADC.ino | Samotné ADC

```
// Analogový snímač teploty - projekt 19
// A0 snímá dělič s termistorem. ADC samotné není teplota.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(1);
}
```

První krok vypisuje pouze hodnotu analogového převodníku. Následující samostatný program používá stejné zapojení a přidává původní převodní rovnici.

Projekt 19 - ukázkový program 2/2

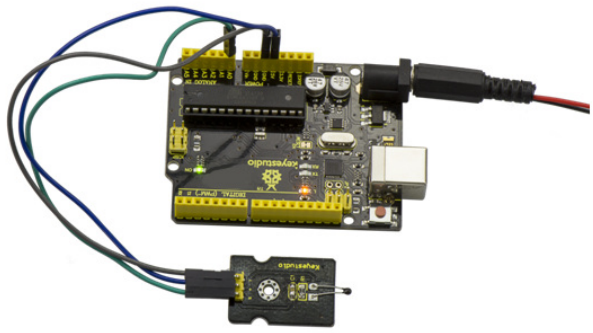
Projekt_19_Teplota.ino | Převod na teplotu

```
// Analogový snímač teploty - projekt 19
// A0 snímá dělič s termistorem. ADC samotné není teplota.
// Oprava je popsána u projektu v českém návodu.
// Druhá ukázka: B=3950 K, NTC 10 kΩ při 25 °C.

#include <math.h>
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  double val = analogRead(0);
  if (val <= 0 || val >= 1023) {
    Serial.println("Chyba: krajni hodnota ADC");
    // Prodleva je v milisekundách: 1000 ms = 1 s.
    delay(1000);
    return;
  }
  // ADC převedeme na napětí při referenci 5 V.
  double fenya = (val / 1023) * 5;
  // Z napětí vypočítáme odpor NTC v děliči.
  double r = (5 - fenya) / fenya * 4700;
  // B rovnice: odpor převedeme na kelviny a poté na °C.
  Serial.println( 1 / ( log(r / 10000) / 3950 + 1 / (25 + 273.15)) - 273.15);
  delay(1000);
}
```

Očekávaný výsledek

První program vypisuje číslo 0 až 1023. Druhý vypočítává teplotu ve stupních Celsia. Zahřátí snímací části prsty má způsobit postupnou změnu; referenční teploměr pomůže posoudit odchylku.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Zaznamenejte údaj při pokojové teplotě a po opatrném ohřátí prsty. Změna konstant bez znalosti skutečného termistoru nenahrazuje kalibraci.

Projekt 20: Otočný potenciometr

Popis



Otočný ovladač je potenciometr zapojený jako dělič napětí. Poloze hřídele odpovídá analogové napětí na signálovém vývodu. Desetibitový převodník UNO jej při obvyklém nastavení převede na jednu z 1024 hodnot. Takto lze zadávat jas, hlasitost nebo požadovanou úroveň jiné veličiny.

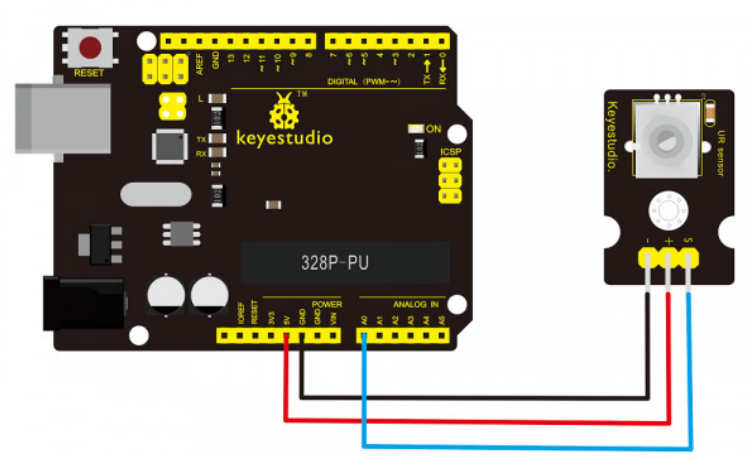
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: analogové
- Rozměry: 30 × 20 mm
- Hmotnost: 8 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Jezdec potenciometru vybírá část napájecího napětí. ADC z něj vytvoří číslo, které může program později využít například jako požadovaný jas.

První ověření a častá chyba

Při otáčení očekávejte plynulý trend. Několik jednotek kolísání u zastavené polohy je běžných. Koncové hodnoty ani střed nemusí být dokonale přesné.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 20 - ukázkový program

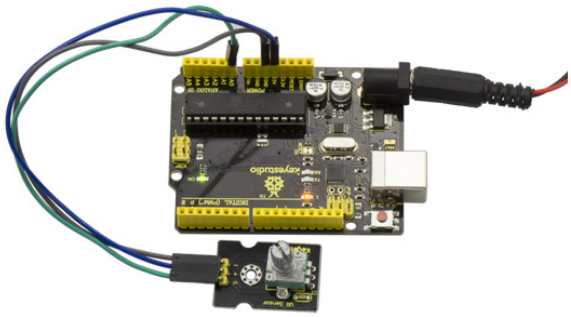
Projekt_20.ino

```
// Otočný potenciometr - projekt 20
// Hodnota A0 roste nebo klesá podle natočení potenciometru.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int val;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  val = analogRead(0);
  Serial.println(val, DEC);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(100);
}
```

Očekávaný výsledek

Při 9600 baud sledujte hodnoty v Sériovém monitoru. Otáčením se mění přibližně od 0 do 1023. Mechanickými dorazy neotáčejte silou.

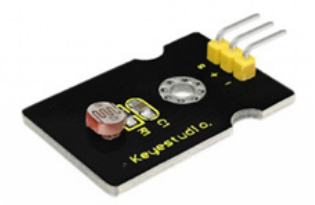


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Jako další pokus převeďte hodnotu 0 až 1023 na PWM 0 až 255 a řiďte jí jas LED. Nejprve ale ověřte samotný potenciometr tímto základním programem.

Projekt 21: Fotorezistor

Popis



Fotorezistor mění svůj elektrický odpor podle osvětlení. Modul převádí tuto změnu na analogové napětí. Podobné snímání lze použít pro jednoduché soumrakové spínání, reakci na zastínění nebo sledování světelných podmínek. Číselný výstup tohoto základního programu není kalibrovaný údaj v luxech.

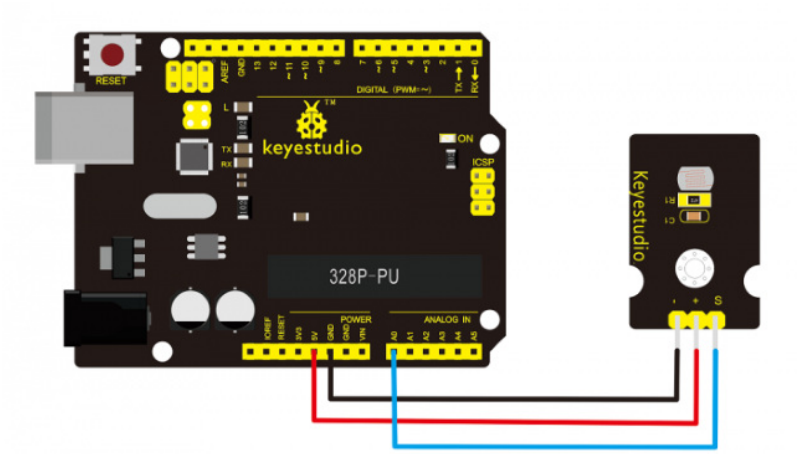
Parametry

- Napájení: 5 V
- Rozhraní: analogové
- Rozměry: 30 × 20 mm
- Hmotnost: 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Změna odporu fotorezistoru změní poměr v děliči. Směr změny čísla závisí na tom, v které větvi děliče je snímač zapojen.

První ověření a častá chyba

Zapište hodnotu při osvětlení a při zakrytí za stejných podmínek. Práh pro soumrak potom zvolte mezi nimi, nikoli podle univerzálního čísla z jiného stolu.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 21 - ukázkový program

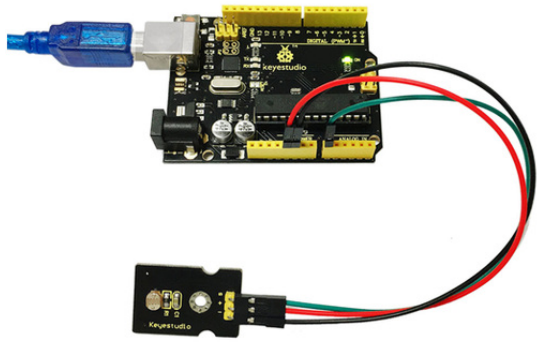
Projekt_21.ino

```
// Fotorezistor - projekt 21
// Zastínění fotorezistoru mění napětí na A0.

int sensorPin = A0 ;
int value = 0;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  value = analogRead(sensorPin);
  Serial.println(value, DEC);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(50);
}
```

Očekávaný výsledek

V Sériovém monitoru při 9600 baud přibývá hodnota každých 50 ms. U modulu z předlohy při zakrytí rukou údaj klesá.

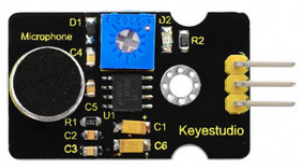


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Změřte hodnotu v běžném světle a pod krabičkou. Pro budoucí světelný spínač zvolte dva různé prahy pro zapnutí a vypnutí, aby světlo neblikalo na hranici rozhodnutí.

Projekt 22: Analogový snímač zvuku

Popis



Mikrofon s doplňující elektronikou převádí zvuk na elektrický signál. Modul může posloužit jako vstup pro reakci na hlas nebo tlesknutí. Základní ukázka předlohy vypisuje jednotlivé odečty z analogového vstupu. Je vhodná k prvnímu pozorování, ale neprovádí kalibrované měření hlasitosti.

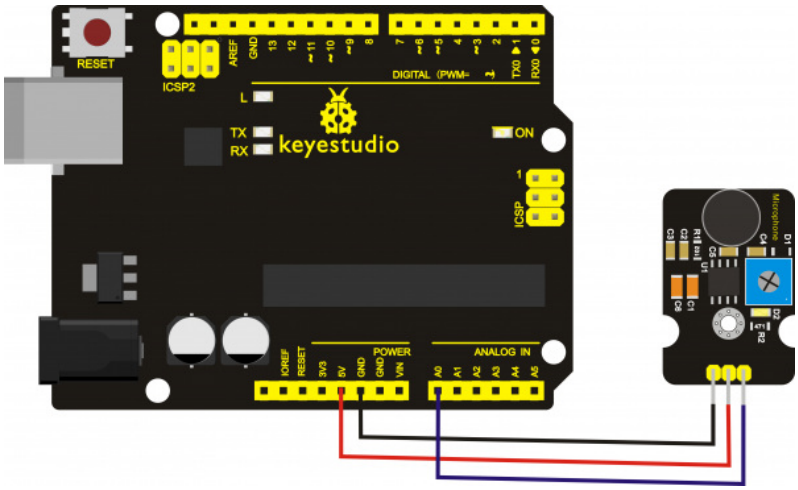
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: analogové
- Rozměry: 30 × 20 mm
- Hmotnost: 4 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava textu HWKITCHEN: v předloze se pod zapojením opakuje obecný popis. Zde je výslovně uvedeno S → A0, + → 5 V a - → GND podle původního schématu. Hodnota ADC není údaj v decibelech.

Jak pokus funguje

Zvuk je rychle proměnný průběh. Jeden vzorek po 100 ms jej nepopisuje celý a může minout tlesknutí. Pro detekci zvukové události je vhodnější sledovat více rychlých vzorků v časovém okně.

První ověření a častá chyba

Porovnejte klid a déle trvající zvuk. Pokud jednotlivá čísla nevypadají jednoznačně, neznamená to samo o sobě vadný mikrofon. Otvor mikrofonu nezakrývejte kapalinou ani do něj nefoukejte zblízka.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 22 - ukázkový program

Projekt_22.ino

```
// Analogový snímač zvuku - projekt 22
// Jednotlivý vzorek zvuku je číslo ADC, nikoli decibely.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int val;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  val = analogRead(0);
  Serial.println(val, DEC);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(100);
}
```

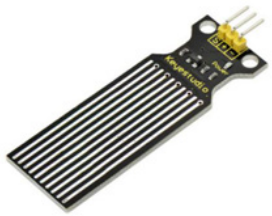
Očekávaný výsledek

Otevřete Sériový monitor na 9600 baud. Mluvte směrem k mikrofonu a sledujte změny čísel. Jednotlivý odečet po 100 ms nemusí zachytit každý krátký zvuk.

Tip HWKITCHEN - vyzkoušejte navíc
Jako rozšíření vyhodnocujte během krátkého časového okna nejnižší a nejvyšší vzorek a vypište jejich rozdíl. Porovnejte jej se základním jednorázovým měřením.

Projekt 23: Snímač hladiny vody

Popis



Na měřicí části jsou odkryté vodivé dráhy. Voda mezi nimi vytvoří vodivé cesty a změní analogový výstup. Senzor tak může rozlišovat různé úrovně smočení nebo přítomnost kapek. Program z předlohy současně rozsvěcí vestavěnou LED, pokud hodnota přesáhne práh 700.

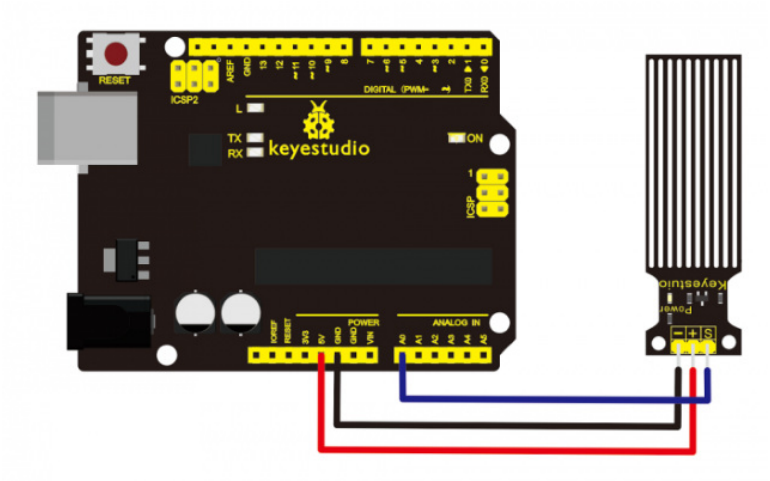
Parametry

- Napájení: 5 V DC
- Proud: méně než 20 mA
- Rozhraní: analogové
- Měřicí plocha: 40 × 16 mm
- Deska: FR4 s pocínovanými vodivými drahami
- Teplota okolí dle předlohy: 10 až 30 °C
- Vlhkost okolí dle předlohy: 10 až 90 % bez kondenzace
- Rozměry: 65 × 20 × 8 mm; hmotnost 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Proud mezi mokrými drahami závisí na vodivosti vody i ploše smočení. Podmínka > 700 je demonstrační práh; program sám nepřevádí hodnotu na centimetry hladiny.

První ověření a častá chyba

Změřte suchý stav a několik hloubek ve stejné vodě. Destilovaná voda může reagovat slabě. Po pokusu odpojte napájení, plochu očistěte a osušte.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 23 - ukázkový program

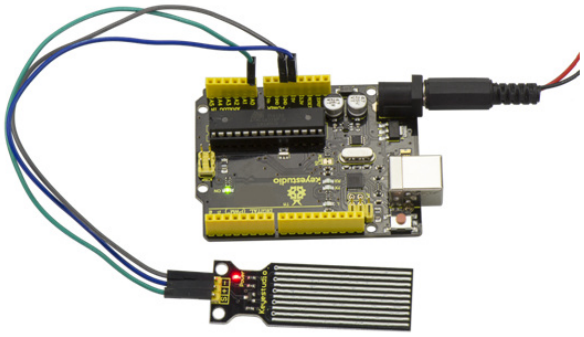
Projekt_23.ino

```
// Snímač hladiny vody - projekt 23
// Hodnota nad 700 rozsvítí D13; práh pochází z předlohy.

int analogPin = 0;
int led = 13;
int val = 0;
int data = 0;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(led, OUTPUT);
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  val = analogRead(analogPin);
  if (val > 700) {
    digitalWrite(led, HIGH);
  }
  else {
    digitalWrite(led, LOW);
  }
  data = val;
  Serial.println(data);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(100);
}
```

Očekávaný výsledek

Ponořujte jen určenou měřicí plochu do malé nádobky. Při 9600 baud sledujte měnící se čísla. Pokud hodnota překročí 700, rozsvítí se D13; jinak zhasne. Práh nemusí odpovídat každému typu vody.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Změřte suchý stav a několik hloubek ponoření. Vyberte vlastní práh a teprve potom případně přidejte zvukový alarm. Konektor a horní elektroniku udržujte suché, po pokusu modul osušte.

Projekt 24: Snímač vlhkosti půdy

Popis



Dvě vodivé sondy měří elektrickou vodivost materiálu mezi nimi. Vlhčí půda bývá vodivější, takže se výstup obvykle zvyšuje. Výsledek ovlivňují také minerály, složení zeminy, hloubka zasunutí a kontakt se sondami. Princip se hodí k orientačnímu sledování zálivky; základní ukázka vypisuje analogovou hodnotu.

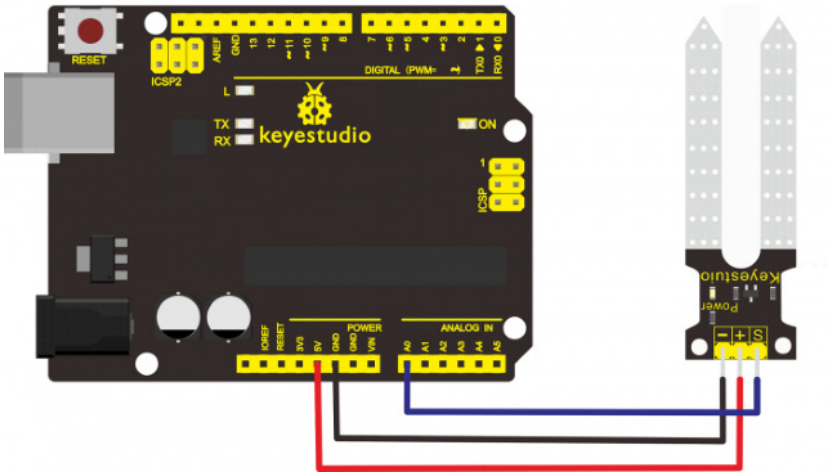
Parametry

- Napájení: 3,3 nebo 5 V
- Proud: nejvýše 20 mA
- Analogový výstup dle předlohy: přibližně 0 až 2,3 V
- Vývody: signál, GND, VCC; řiďte se potiskem
- Rozměry: 20 × 60 mm
- Povrch sond je upraven pro lepší vodivost a odolnost

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Upřesnění HWKITCHEN: údaj „životnost asi jeden rok“ z předlohy závisí na prostředí a způsobu provozu. Pásma hodnot v ukázce navíc nemusí souhlasit s uvedeným napětím 2,3 V; rozhodující je vlastní měření, nikoli odhad procent vlhkosti.

Jak pokus funguje

Odporový snímač reaguje na vodivost půdy, nikoli přímo na procento vody. Stejné číslo v jiné zemině proto nemusí znamenat stejnou potřebu zálivky.

První ověření a častá chyba

Použijte stejnou hloubku sondy a porovnejte suchou a zalitou zeminu. Napájená mokrá sonda koroduje; po krátkém pokusu ji odpojte a očistěte.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 24 - ukázkový program

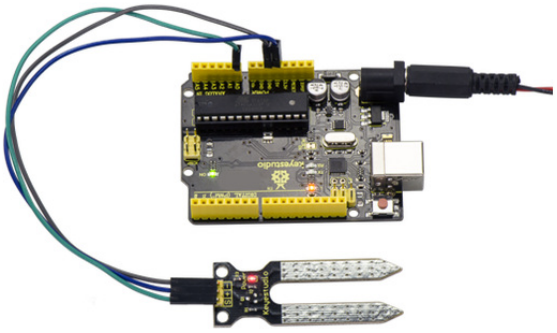
Projekt_24.ino

```
// Snímač vlhkosti půdy - projekt 24
// Pozor: tento původní program používá rychlost 57600 baud.
// Pásma 0–300 / 300–700 / 700–950 jsou pouze orientační.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 57600 baud nastavte v Sériovém monitoru.
  Serial.begin(57600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  Serial.print("Moisture Sensor Value:");
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  Serial.println(analogRead(0));
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(100);
}
```

Očekávaný výsledek

Nastavte monitor na 57600 baud. Vypisuje se text „Moisture Sensor Value:“ a číslo, které se mění podle půdy. Předloha uvádí orientační pásma 0-300, 300-700 a 700-950; nepovažujte je za univerzální kalibraci.

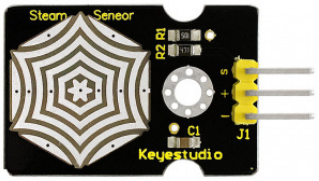


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Změřte stejnou zeminu suchou a po zalití, vždy při stejné hloubce zasunutí. Pro dlouhodobé měření omezte dobu napájení sond, aby se zpomalila elektrochemická koroze.

Projekt 25: Snímač kapek a kondenzace

Popis



Modul označený v předloze jako Steam Moisture reaguje na vodu na vodivém povrchu. Je použitelný pro jednoduchou detekci kapek, deště nebo kondenzace. Nejde o běžný snímač relativní vlhkosti vzduchu. Konektor a připojovací část nejsou vodotěsné.

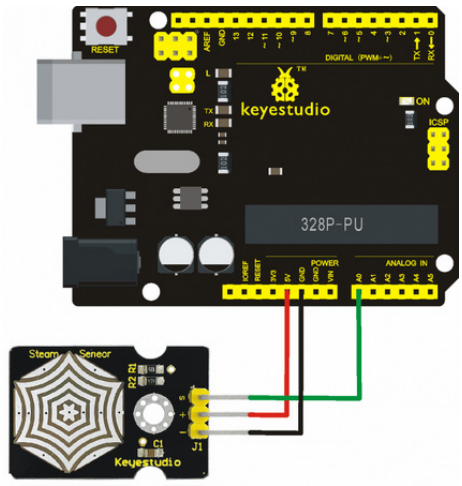
Parametry

- Napájení: 3,3 nebo 5 V
- Proud: méně než 20 mA
- Teplota okolí: -10 až +70 °C
- Rozhraní: analogové
- Rozměry: 36 × 20 mm
- S = signál, + = napájení, - = zem

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Vodivé spoje na ploše propojí kapka. Modul rozpoznává smočení, zatímco relativní vlhkost vzduchu v procentech měří až DHT11.

První ověření a častá chyba

Vyzkoušejte jednu kapku a její setření. Horká pára by mohla poškodit modul i opařit obsluhu; ke zkoušce stačí voda pokojové teploty.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 25 - ukázkový program

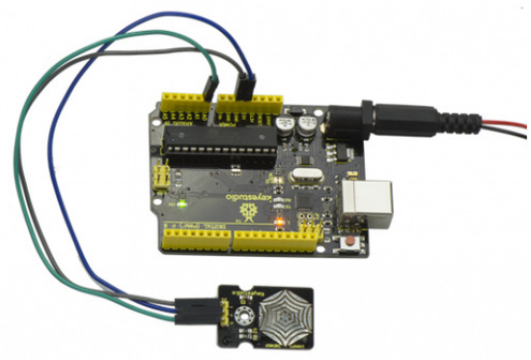
Projekt_25.ino

```
// Snímač kapek a kondenzace - projekt 25
// Voda na měřicí ploše mění napětí; konektor musí být suchý.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int val;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  val = analogRead(0);
  Serial.print("Moisture is ");
  Serial.println(val, DEC);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(100);
}
```

Očekávaný výsledek

Při 9600 baud sledujte řádky „Moisture is …“. Kápněte malé množství vody jen na měřicí plochu a porovnejte čísla se suchým stavem.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Zkuste jednu kapku, více kapek a následné osušení. Použijte vodu pokojové teploty; horká pára není pro tento pokus potřeba.

Projekt 26: Keramický piezoelektrický snímač vibrací

Popis



Piezoelektrický keramický prvek při mechanickém namáhání vytváří elektrický signál. Přes modul lze tento signál pozorovat na analogovém vstupu. Předloha používá samostatné odečty a výpis do monitoru. Při různě silném jemném poklepání se mění naměřená hodnota; velmi krátký impuls může mezi odečty uniknout.

Parametry

- Výstup: analogový
- Teplota okolí dle předlohy: -10 až +70 °C
- Hmotnost: 4,4 g
- Předloha obecně uvádí 3,3 až 5 V a méně než 1 mA; zobrazený vývod NC není napájení

Zapojení

UNO, USB kabel, keramický vibrační modul a 2 vodiče. Vývod NC zůstává volný.

Modul	Arduino / spoj
S	A0
- / G	GND
NC	nezapojovat

Oprava nebo upřesnění předlohy
Upřesnění HWKITCHEN: NC je u tohoto konkrétního modulu nevyužitý vývod. Připojte S na A0 a - na GND. Původní obrázek vede 5 V na NC; tento zbytečný vodič není potřeba. Zde proto používáme jednoznačnou tabulku zapojení. Nejde o kontakt NC jako u relé.

Jak pokus funguje

Piezoelektrický prvek reaguje především na změnu mechanického namáhání. Krátký impuls se může odehrát mezi dvěma pomalými odečty; program proto není spolehlivým počítadlem všech úderů.

První ověření a častá chyba

Použijte hotový modul a jemný pohyb podložky. Holý piezoelektrický prvek může vytvořit vyšší napětí a není automaticky bezpečnou přímou náhradou na vstupu UNO.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 26 - ukázkový program

Projekt_26.ino

```
// Keramický piezoelektrický snímač vibrací - projekt 26
// Čteme krátké piezoelektrické impulsy. NC nezapojujte.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int val;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  val = analogRead(0);
  Serial.print("Vibration is ");
  Serial.println(val, DEC);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(100);
}
```

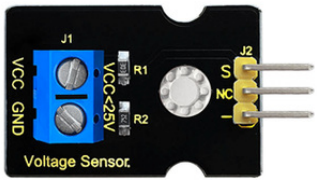
Očekávaný výsledek

V monitoru na 9600 baud sledujte „Vibration is …“. Opatrně rozvibrujte keramický kotouček nebo podložku, na které leží. Prvek neohýbejte a nebouchejte do něj tvrdým předmětem.

Tip HWKITCHEN - vyzkoušejte navíc
Jako doplnění zkuste hledat maximum z více rychlých odečtů. Uvidíte, proč jednoduché čtení jednou za 100 ms některé klepnutí nezachytí.

Projekt 27: Snímač napětí

Popis



Modul je odporový dělič, který zmenší vstupní napětí přibližně pětkrát. Analogový vstup UNO pak měří zmenšenou hodnotu. Pro referenci 5 V odpovídá teoretický horní rozsah děliče 25 V, pro 3,3V systém 16,5 V. Jeden krok desetibitového převodníku při referenci 5 V představuje na vstupu děliče přibližně 24,4 mV. Pro první pokus použijte pouze jeden bateriový článek.

Parametry

- Vstupní rozsah dle předlohy: 0 až 25 V DC při 5V systému
- Dělicí poměr: přibližně 5 : 1
- Výstup: analogový
- Rozměry: 35 × 20 × 14 mm
- Svorky J1: VCC a GND pro měřené napětí
- Výstup J2: S, nezapojený NC, zem

Zapojení

UNO, USB kabel, napěťový modul, dva signálové vodiče a jeden bateriový článek s bezpečným držákem pro připojení do svorkovnice.

Modul	Arduino / spoj
J2: S	A0
J2: - / G	GND
J2: NC	nezapojovat
J1: VCC	plus měřené baterie
J1: GND	minus měřené baterie

Oprava nebo upřesnění předlohy
Opravy HWKITCHEN: NC u tohoto modulu zůstává volný; původní obrázek s vodičem 5 V na NC nahrazuje tabulka. Převod val / 3.9 nahrazuje val × 250 / 1023: přibližné desetiny voltu při děliči 5 : 1 a referenci 5 V. Nejde o kalibraci. Formát výpisu zůstává. Neměřte síťové ani záporné napětí a neznámé zdroje.

Jak pokus funguje

ADC měří zmenšené napětí, program je násobí poměrem děliče. Výsledek ovlivní skutečné napájecí napětí UNO i tolerance rezistorů. Nominálních 25 V není doporučené zkušební napětí ani bezpečnostní rezerva.

První ověření a častá chyba

Začněte jedním 1,5V článkem a porovnejte údaj s multimetrem. Měřený obvod je přes GND spojený s Arduinem i USB počítače; dělič neposkytuje galvanické oddělení.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 27 - ukázkový program

Projekt_27.ino

```
// Snímač napětí - projekt 27
// Dělič zmenšuje měřené napětí pětkrát. NC nezapojujte.
// val1 a val5 vyjadřují desetiny voltu, ne celé volty.
// Oprava je popsána u projektu v českém návodu.

int analogpin = 0;
int val, val5;
int val2 = 0;
int val3 = 0;
int val4 = 0;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int val, val5;
  float val1;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  val = analogRead(analogpin);
  // Převod ADC na desetiny voltu: 25 V × 10 / 1023.
  val1 = val * 250.0 / 1023.0;
  // Převedení na celé desetiny, stejně jako v předloze.
  val5 = (int)val1;
  // Desítky voltů.
  val3 = val5 / 100;
  // Jednotky voltů.
  val2 = (val5 % 100) / 10;
  // Desetiny voltu.
  val4 = val5 % 10;
  // Původní řídicí řetězce se v monitoru zobrazí jako text.
  Serial.print("$CLEAR\r\n");
  Serial.print("$GO 1 1\r\n");
  Serial.print("$PRINT Voltage:\r\n");
  Serial.print("$GO 1 9\r\n");
  Serial.print("$PRINT ");
  Serial.print(val3);
  Serial.print(val2);
  Serial.print(".");
  Serial.print(val4);
  Serial.println("V");
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(1000);
}
```

Očekávaný výsledek

Na svorkovnici J1 připojte plus a minus nízkonapěťové baterie se správnou polaritou. Monitor na 9600 baud vypisuje číselnou hodnotu napětí. Zachované řetězce $CLEAR, $GO a $PRINT jsou příkazy původního zobrazovacího řešení; v běžném Sériovém monitoru se zobrazí jako text.

Tip HWKITCHEN - vyzkoušejte navíc
Porovnejte výsledek s multimetrem. Přesnost omezuje reference 5 V, tolerance děliče a zaokrouhlení původního programu na desetiny voltu. Nezaměňujte společnou zem za galvanické oddělení.

Projekt 28: Infračervený přijímač

Popis



Přijímací modul rozpoznává modulovaný infračervený signál dálkového ovladače. Elektrické impulsy následně dekóduje knihovna IRremote. Předloha používá starší rozhraní s objektem IRrecv a strukturou decode_results; pro zachování programu používáme odpovídající řadu knihovny 2.x.

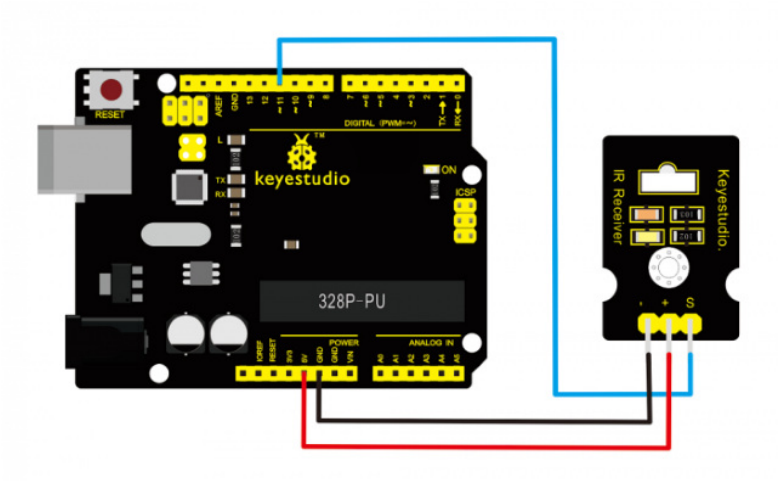
Parametry

- Napájení: 5 V
- Rozhraní: digitální
- Nosná frekvence přijímače: 38 kHz
- Rozměry: 30 × 20 mm
- Hmotnost: 4 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D11
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Infračervené pulzy jsou kódované podle protokolu. Knihovna z nich sestaví zprávu a program ji vypíše v šestnáctkové soustavě (HEX). Stejné tlačítko může při držení posílat opakovací zprávu.

První ověření a častá chyba

Namiřte běžný IR ovladač, nikoli ovladač používající pouze Bluetooth. Srovnávejte krátké stisky stejného tlačítka. Silné okolní světlo může příjem zhoršit.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 28 - ukázkový program

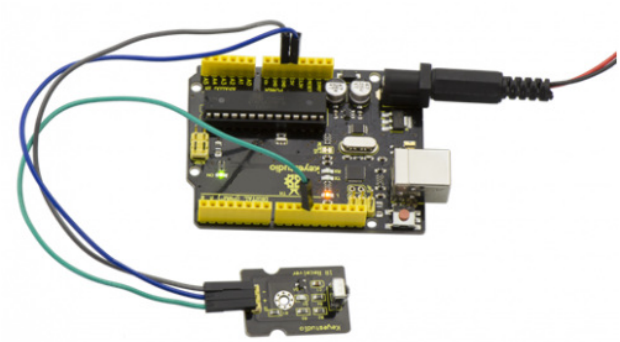
Projekt_28.ino

```
// Infračervený přijímač - projekt 28
// Původní rozhraní IRremote 2.x; signál přijímače je na D11.

#include <IRremote.h>
int RECV_PIN = 11;
// Objekt přijímače používá výše zvolený pin.
IRrecv irrecv(RECV_PIN);
// Sem knihovna uloží dekódovanou zprávu.
decode_results results;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
  irrecv.enableIRIn();
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Pokračujeme jen po přijetí celé rozpoznané zprávy.
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    // Připravíme přijímač na další zprávu.
    irrecv.resume();
  }
}
```

Očekávaný výsledek

Otevřete monitor na 9600 baud a namiřte na snímač IR ovladač. Rozpoznané zprávy se vypisují jako šestnáctková čísla. Kontrolka modulu může při příjmu blikat.

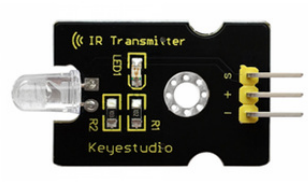


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Zapište kódy dvou tlačítek a opakujte měření. Dlouhý stisk může posílat opakovací zprávy odlišné od prvního stisku. Bluetooth ovladač bez IR vysílače tento pokus neobslouží.

Projekt 29: Infračervený vysílač

Popis



Vysílací modul posílá infračervené světlo k přijímači. Přenos nejlépe funguje při přímém výhledu; odrazy a úhel natočení mohou dosah změnit. Kapitola zachovává dva kroky předlohy: jednoduché blikání výstupu a potom komunikaci dvojice UNO s IR vysílačem a přijímačem. Pro komunikační část jsou níže dva samostatné programy.

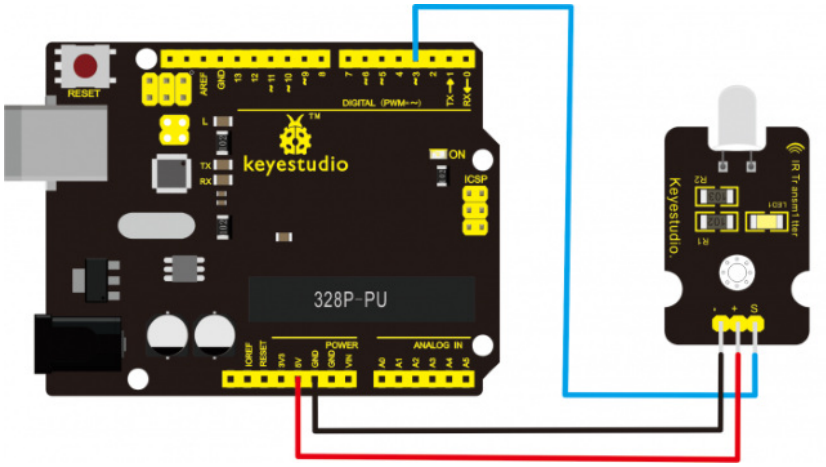
Parametry

- Napájení: 3 až 5 V
- Vlnová délka uváděná předlohou: 850 až 940 nm
- Vyzařovací úhel: přibližně 20°
- Dosah dle předlohy: přibližně 1,3 m při 5 V a modulaci 38 kHz
- Montážní otvory: průměr 3,2 mm, rozteč 15 mm
- Rozměry: 35 × 20 mm; hmotnost 3 g

Zapojení

Základní pokus: UNO, USB kabel, IR vysílač a 3 vodiče. Komunikace: dvě desky UNO, dva datové USB kabely, IR vysílač, IR přijímač a 6 vodičů.

Modul	Arduino / spoj
S	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Opravy HWKITCHEN: vysílač patří na D3 a přijímač na D11. Komunikace vyžaduje dvě desky. Do přijímače je doplněno pinMode(LED_PIN, OUTPUT). Vysílač nyní předává 12 datových bitů RC5 místo zkrácených 8: jeden toggle bit, pět bitů adresy a šest bitů příkazu. Startovní bity doplní původní rozhraní IRremote. Hodnoty 0/1 a prodlevy 200 ms zůstávají; nejde o úplnou emulaci ovladače s přepínáním toggle bitu.

Jak pokus funguje

Pomalé blikání je první zkouška výstupu, nepřenáší ještě kódovanou zprávu. Ve druhé části knihovna vytváří nosnou a rámec RC5; druhé UNO jej dekóduje a řídí LED.

První ověření a častá chyba

Na každé UNO nahrajte správnou roli. Nejprve ověřte přijímač projektem 28. Obě desky používají IRremote 2.x; optický přenos mezi nimi nepotřebuje společný signálový vodič.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 29 - ukázkový program 1/3

Projekt_29_Blikani.ino | Jednoduché blikání

```
// Infračervený vysílač - projekt 29
// Vysílač používá D3. Přijímač na druhém UNO používá D11.

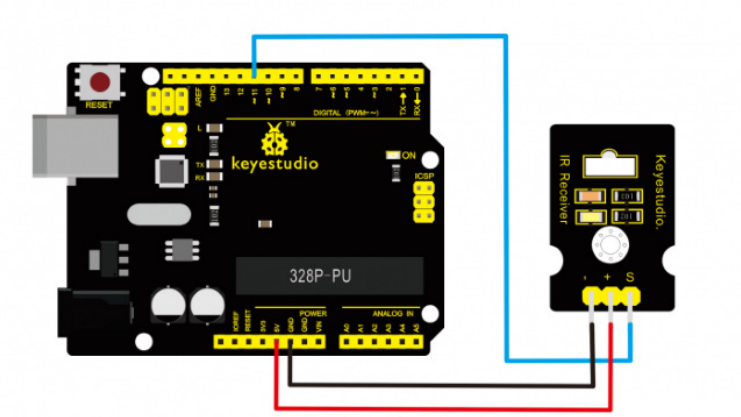
int led = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(led, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  digitalWrite(led, HIGH);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Výsledek prvního kroku

Indikační LED vysílače střídá sekundu svícení a sekundu zhasnutí. Pro kódovanou komunikaci pokračujte dvojicí programů na dalších stranách.

Komunikační část - dvě desky UNO

Vysílač: S → D3, + → 5 V, - → GND. Přijímač na druhém UNO: S → D11, + → 5 V, - → GND. Každá deska potřebuje svůj příslušný program a napájení.



Originální schéma přijímače pro komunikační část: Keyestudio.

Projekt 29 - ukázkový program 2/3

Projekt_29_Vysilac.ino | Odesílající UNO

```
// Infračervený vysílač - projekt 29
// Vysílač používá D3. Přijímač na druhém UNO používá D11.
// Oprava je popsána u projektu v českém návodu.

#include <IRremote.h>
// Objekt vysílače: stará knihovna na UNO vysílá na D3.
IRsend irsend;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // RC5: 12 datových bitů; start přidává knihovna.
  irsend.sendRC5(0x0, 12);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(200);
  // Odeslání hodnoty 1 ve stejném formátu.
  irsend.sendRC5(0x1, 12);
  delay(200);
}
```

Projekt 29 - ukázkový program 3/3

Projekt_29_Prijimac.ino | Přijímající UNO

```
// Infračervený vysílač - projekt 29
// Vysílač používá D3. Přijímač na druhém UNO používá D11.
// Oprava je popsána u projektu v českém návodu.

#include <IRremote.h>
const int RECV_PIN = 11;
const int LED_PIN = 13;
IRrecv irrecv(RECV_PIN);
decode_results results;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
  // Řídicí pin nastavíme jako výstup.
  pinMode(LED_PIN, OUTPUT);
  irrecv.enableIRIn();
}
// loop() Arduino stále znovu opakuje.
void loop() {
  if (irrecv.decode(&results)) {
    if ( results.bits > 0 ) {
      int state;
      // Přijatá hodnota 1 má rozsvítit vestavěnou LED.
      if ( 0x1 == results.value ) {
        state = HIGH;
      }
      else {
        state = LOW;
      }
      digitalWrite( LED_PIN, state );
    }
    // Po zpracování znovu povolíme příjem.
    irrecv.resume();
  }
}
```

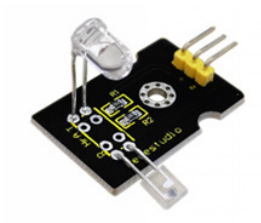
Očekávaný výsledek

První program střídá jednu sekundu zapnuto a jednu sekundu vypnuto; indikační LED ukazuje aktivitu. V komunikační části vysílač posílá dvě hodnoty střídavě po 200 ms. Přijímač při rozpoznané hodnotě 1 rozsvítí D13 a při jiné hodnotě ji zhasne.

Tip HWKITCHEN - vyzkoušejte navíc
Měňte vzdálenost a úhel mezi vysílačem a přijímačem, vždy pouze jednu věc. Při potížích nejprve ověřte přijímač podle projektu 28 a zvolenou verzi knihovny. Pomalé blikání z první ukázky není kódovaná zpráva pro přijímač dálkového ovládání.

Projekt 30: Optický snímač pulzu

Popis



Světelný zdroj a fototranzistor sledují změny procházejícího nebo odraženého světla při přiložení prstu. Drobné změny mohou souviset s prokrvením. Původní program vypisuje surový signál a jednoduchý exponenciální filtr. Zachováváme koeficient alpha = 0,75 a interval 20 ms.

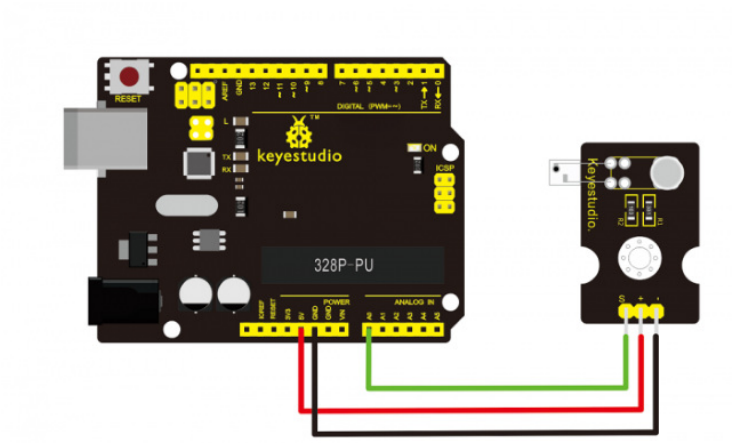
Parametry

- Napájení v ukázce: 5 V
- Signál: analogový na A0
- Výpis: 115200 baud
- Vzorkovací prodleva: 20 ms
- Výstup programu: surová a filtrovaná hodnota

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Filtr kombinuje 75 % předchozího výsledku a 25 % nového vzorku. Tlumí rychlé změny, ale také zpožďuje odezvu. Zde se nevyhledávají jednotlivé tepy ani nepočítá BPM.

První ověření a častá chyba

Prst přiložte volně a zůstaňte v klidu. Silný přítlak a pohyb mohou signál změnit více než pulz. Jde o experimentální snímač, ne zdravotnické měření.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 30 - ukázkový program

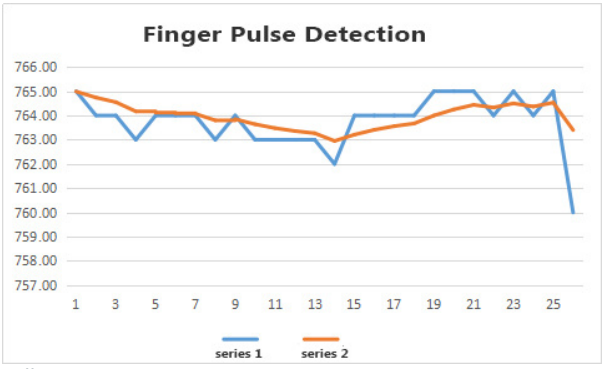
Projekt_30.ino

```
// Optický snímač pulzu - projekt 30
// Filtr: 75 % staré hodnoty + 25 % nového vzorku.
// Vypisujeme dva průběhy; tento program nepočítá tep za minutu.

int ledPin = 13;
int sensorPin = 0;
// Váha předchozího výsledku filtru.
double alpha = 0.75;
// Prodleva mezi odečty v milisekundách.
int period = 20;
double change = 0.0;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(ledPin, OUTPUT);
  // Stejných 115200 baud nastavte v Sériovém monitoru.
  Serial.begin(115200);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // static uchovává hodnotu mezi průchody loop().
  static double oldValue = 0;
  static double oldChange = 0;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  int rawValue = analogRead(sensorPin);
  // Exponenciální filtr tlumí rychlé změny a šum.
  double value = alpha * oldValue + (1 - alpha) * rawValue;
  Serial.print(rawValue);
  Serial.print(",");
  Serial.println(value);
  // Uložíme výsledek pro příští výpočet.
  oldValue = value;
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(period);
}
```

Očekávaný výsledek

Přiložte prst zlehka a omezte pohyb. Při 115200 baud jsou v každém řádku dvě čísla oddělená čárkou. Lze je zakreslit nebo přenést do tabulky, jak ukazuje předloha. Samotný program nepočítá tepovou frekvenci za minutu.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Porovnejte průběh při klidném a pohybujícím se prstu. Změňte alpha například na 0,9: filtr více vyhlazuje, ale reaguje pomaleji. Pomůcka slouží k pokusům se signálem, nikoli ke zdravotnímu posouzení.

Projekt 31: Joystick

Popis



Dva potenciometry snímají vychýlení páčky ve směru X a Y. Zatlačení na páčku navíc ovládá tlačítko. Modul proto potřebuje dva analogové vstupy a jeden digitální vstup. Ukázka předlohy vypisuje všechny tři hodnoty v každém řádku.

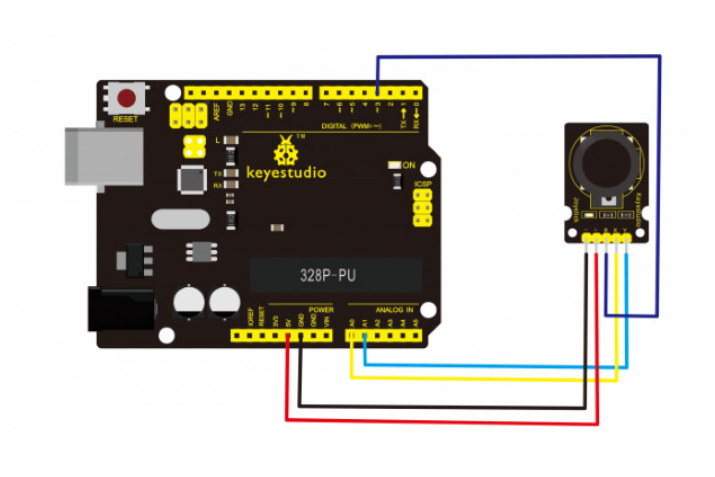
Parametry

- Napájení: 3,3 až 5 V
- Rozhraní: 2× analogové a 1× digitální
- Rozměry: 40 × 28 mm
- Hmotnost: 12 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 5 propojovací vodiče.

Modul	Arduino / spoj
X	A0
Y	A1
B / SW	D3
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Jak pokus funguje

Osy X a Y jsou dva nezávislé analogové kanály. Tlačítko Z je digitální. Pro řízení robota by bylo vhodné kolem středu zavést malou necitlivou oblast.

První ověření a častá chyba

Nejdříve si zapište střed a oba krajní směry každé osy. Pokud tlačítko v klidu náhodně kolísá, ověřte pull-up konkrétního modulu; INPUT_PULLUP lze použít, když tlačítko skutečně spíná proti GND.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 31 - ukázkový program

Projekt_31.ino

```
// Joystick - projekt 31
// X je A0, Y je A1 a tlačítko páčky je na D3.

int JoyStick_X = 0;
int JoyStick_Y = 1;
int JoyStick_Z = 3;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Senzorový pin bude vstupem.
  pinMode(JoyStick_Z, INPUT);
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int x, y, z;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  x = analogRead(JoyStick_X);
  y = analogRead(JoyStick_Y);
  z = digitalRead(JoyStick_Z);
  Serial.print(x , DEC);
  Serial.print(",");
  Serial.print(y , DEC);
  Serial.print(",");
  Serial.println(z , DEC);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(100);
}
```

Očekávaný výsledek

Při 9600 baud sledujte hodnoty X,Y,Z. Vychýlením páčky se mění první dvě čísla; stiskem se mění třetí. Střed nemusí být přesně 512. Pozorujte směry nahoru, dolů, vlevo a vpravo jako v předloze.

Původní dokumentace ukazuje odečty pro střed a čtyři směry pohybu. Přesné hodnoty se u vašeho kusu mohou lišit; sledujte především změnu os X a Y.

Tip HWKITCHEN - vyzkoušejte navíc
Změřte klidový rozsah obou os a vytvořte kolem něj malé neutrální pásmo. Tím můžete zabránit samovolnému pomalému pohybu robota při uvolněné páčce.

Projekt 32: Rotační enkodér

Popis



Enkodér poskytuje dva časově posunuté digitální signály. Z jejich vzájemného pořadí lze určit směr otáčení a přičítat nebo odečítat kroky. Oproti potenciometru nemá omezený rozsah otáčení. Předloha používá dvě externí LED k indikaci směru a tlačítko pro vynulování počítadla; toto zapojení i funkce jsou zachovány.

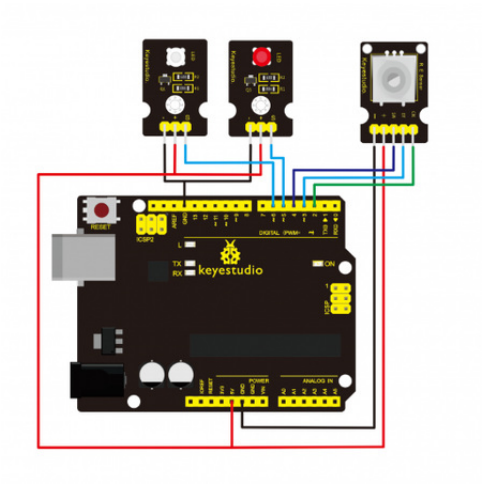
Parametry

- Napájení: 5 V
- Rozhraní: digitální CLK, DAT/DT a tlačítko
- Rozměry: 30 × 20 mm
- Hmotnost: 7 g
- Navíc: dvě LED sestavy, nepájivé pole a vodiče

Zapojení

UNO, USB kabel, enkodér, dva třípinové LED moduly aktivní v HIGH, nepájivé pole a vodiče. Každý LED modul: + na 5 V, - na GND, S na D5 nebo D6. Druhý LED modul může být nutné dokoupit; samotná LED bez rezistoru ani RGB modul nejsou přímou náhradou.

Modul	Arduino / spoj
CLK	D2
DAT / DT	D3
BUTTON / SW	D4
+ / V	5 V
- / G	GND
S LED1 / LED2	D5 / D6



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: delay(200) bylo odstraněno z obsluhy přerušení. Sdílené počítadlo je volatile a jeho čtení či nulování je chráněno krátkým zákazem přerušení. Vstupy a výstupy se nastavují před připojením přerušení. Piny, indikace směru a reset tlačítkem odpovídají předloze.

Jak pokus funguje

Při hraně CLK se čte DAT a podle něj se mění počítadlo. volatile upozorňuje překladač, že proměnnou mění přerušení; atomické čtení brání načtení dvou nesouhlasných bajtů.

První ověření a častá chyba

Otáčejte pomalu. Zákmity mechanického enkodéru mohou způsobit více kroků na jednu aretaci; ukázka není přesný úhloměr. I během dvousekundové prodlevy po resetu mohou přerušení znovu měnit počet.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 32 - ukázkový program

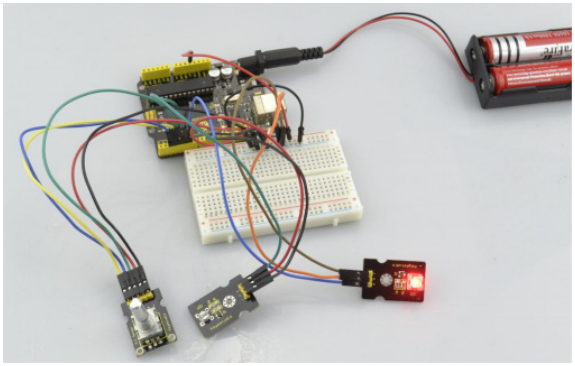
Projekt_32.ino

```
// Rotační enkodér - projekt 32
// CLK=D2, DAT=D3, tlačítko=D4, LED1=D5, LED2=D6.
// Směr se určuje podle DAT při sestupné hraně CLK.
// Oprava je popsána u projektu v českém návodu.

const int interruptA = 0;
const int interruptB = 1;
int CLK = 2;
int DAT = 3;
int BUTTON = 4;
int LED1 = 5;
int LED2 = 6;
// COUNT mění přerušení, proto je označeno volatile.
volatile int COUNT = 0;
void RoteStateChanged();
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Senzorový pin bude vstupem.
  pinMode(CLK, INPUT);
  digitalWrite(2, HIGH);
  pinMode(DAT, INPUT);
  digitalWrite(3, HIGH);
  pinMode(BUTTON, INPUT);
  digitalWrite(4, HIGH);
  // Řídicí pin nastavíme jako výstup.
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
  attachInterrupt(interruptA, RoteStateChanged, FALLING);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Načtení aktuální logické úrovně LOW nebo HIGH.
  if (digitalRead(BUTTON) == LOW) {
    // Krátce ochráníme sdílená data před přerušením.
    noInterrupts();
    COUNT = 0;
    // Znovu povolíme přerušení.
    interrupts();
    Serial.println("STOP COUNT = 0");
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    // Prodleva je v milisekundách: 1000 ms = 1 s.
    delay(2000);
  }
  // Krátce ochráníme sdílená data před přerušením.
  noInterrupts();
  // 16bitový údaj převezmeme při vypnutém přerušení.
  int kopie = COUNT;
  // Znovu povolíme přerušení.
  interrupts();
  Serial.println(kopie);
}
// Obsluha přerušení musí být krátká; nepoužívá delay().
void RoteStateChanged() {
  if (digitalRead(DAT) == HIGH) {
    // Jeden směr: přičteme krok.
    COUNT++;
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
  }
  else {
    // Opačný směr: odečteme krok.
    COUNT--;
    digitalWrite(LED2, HIGH);
    digitalWrite(LED1, LOW);
  }
}
```

Očekávaný výsledek

Při otáčení se mění COUNT vypsané při 9600 baud. Jeden směr rozsvítí LED1 na D5, opačný LED2 na D6. Stisk tlačítka COUNT vynuluje, zhasne obě LED a ponechá původní dvousekundovou prodlevu v hlavní smyčce. Přerušení přitom zůstávají aktivní: otočení během této prodlevy může počítadlo znovu změnit. Jeden mechanický krok nemusí kvůli zákmitům odpovídat jedinému přírůstku.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Mechanický enkodér může při jednom kroku vytvořit více hran. Jako další rozšíření přidejte ošetření zákmitů. Pokud máte jen jednu bílou LED, pro druhý indikátor použijte vhodný jiný LED modul; holá dioda vyžaduje vlastní sériový rezistor.

Projekt 33: Jednokanálové relé

Popis



Relé odděluje řídicí část od přepínaných kontaktů. Tento pokus podle předlohy přes kontakt relé spíná signál LED modulu. Relé se ovládá D8, zatímco D13 je trvale HIGH a přes kontakt přivádí řídicí úroveň do LED modulu. Celá ukázka pracuje pouze s 5 V z Arduina.

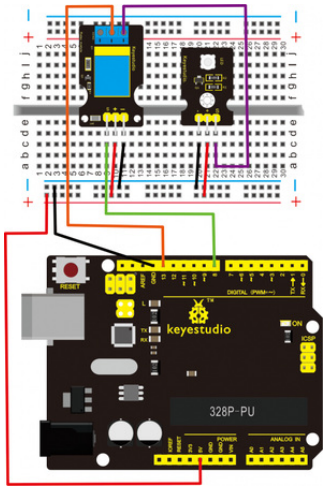
Parametry

- Ovládání: digitální, u zobrazeného modulu aktivní HIGH
- Cívka relé: 5 V
- Kontakty: COM, NO (v klidu rozpojeno), NC (v klidu spojeno)
- Doba přepnutí uváděná předlohou: 10 ms
- Rozměry: 40 × 28 mm; hmotnost 15 g
- Zatížitelnost kontaktů se musí ověřit pro konkrétní relé a desku

Zapojení

UNO, USB kabel, reléový modul, LED modul, nepájivé pole a přibližně 9 propojovacích vodičů. V tomto pokusu není připojena žádná síťová ani výkonová zátěž.

Modul	Arduino / spoj
S relé	D8
+ obou modulů	5 V
- obou modulů	GND
COM relé	D13
NO relé	S LED modulu



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: původní komentáře „Turn off/on“ jsou u aktivního HIGH relé obráceně. Kód se nemění, opravují se komentáře. Tabulka předlohy míchá rozporné údaje 150/250 V AC a různé proudy; nepřebíráme je jako ověřený limit modulu. Tato výuková ukázka není určena pro síťové napětí ani výkonovou zátěž.

Jak pokus funguje

Cívku řídí elektronika modulu, kontakt COM se přepíná mezi NC a NO. NO je v klidu rozpojený; při přitažení relé propojí COM s LED modulem. D13 je zde zdroj logického signálu, nikoli napájení cívky.

První ověření a častá chyba

Poslouchejte cvaknutí a porovnejte je se svícením externí LED. Použijte pouze zapojení s 5 V. NC na relé neznamená nevyužitý pin; je to kontakt sepnutý v klidu.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 33 - ukázkový program

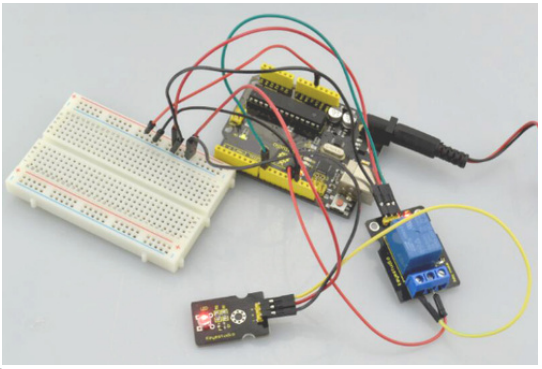
Projekt_33.ino

```
// Jednokanálové relé - projekt 33
// D13 je stále HIGH; přes COM a NO řídí externí LED.
// D8 ovládá cívku: HIGH relé přitáhne, LOW uvolní.

int Relay = 8;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Řídicí pin nastavíme jako výstup.
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
  pinMode(Relay, OUTPUT);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  digitalWrite(Relay, HIGH);
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(2000);
  digitalWrite(Relay, LOW);
  delay(2000);
}
```

Očekávaný výsledek

Relé se na dvě sekundy přitáhne a na dvě sekundy uvolní. Při použití kontaktů COM a NO se externí LED rozsvítí při přitažení. D13 může svítit stále, protože jeho úkolem je zde dodávat trvalý řídicí signál.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
S odpojeným napájením přesuňte vodič z NO na NC a vysvětlete opačné chování LED. Při zapojeném obvodu vodiče nepřepojujte.

Projekt 34: Lineární snímač teploty LM35

Popis



LM35 vytváří napětí úměrné teplotě. Citlivost 10 mV na jeden stupeň Celsia umožňuje jednoduchý lineární výpočet. Předloha používá analogový vstup UNO a vypisuje celé stupně. Tento způsob zobrazení zachováváme, opravujeme pouze typ výpočtu, aby nepřetékal šestnáctibitový int.

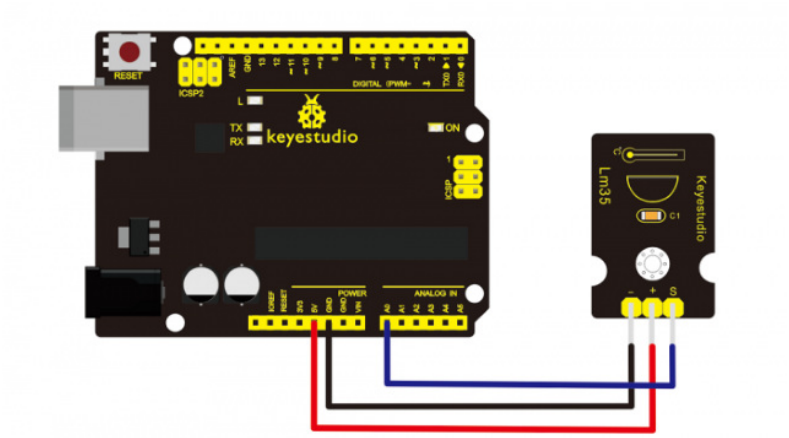
Parametry

- Citlivost: 10 mV/°C
- Rozsah použití modulu uváděný předlohou: 0 až 100 °C
- Napájení v tomto zapojení: 5 V
- Rozhraní: analogové
- Rozměry: 30 × 20 mm; hmotnost 3 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	A0
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: výraz 500 × val je na UNO R3 s 16bitovým int náchylný k přetečení. Změna na 500L × val vynutí výpočet typu long. Dělení 1024, celočíselný výsledek a původní interval zůstávají.

Jak pokus funguje

Při 25 °C odpovídá výstup ideálního LM35 přibližně 0,25 V. Program používá předpokládanou referenci 5 V a celočíselný výpočet, proto na konci zahodí desetinnou část.

První ověření a častá chyba

Porovnejte teplotu po ustálení s teploměrem. Rozlišení ADC, zaokrouhlení dolů a skutečná reference omezují přesnost. Zahřívání prsty je vhodný pomalý test.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 34 - ukázkový program

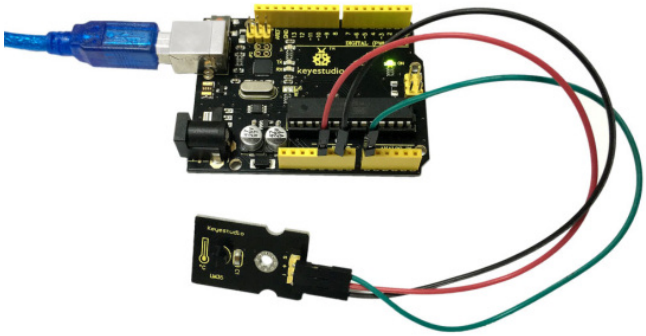
Projekt_34.ino

```
// Lineární snímač teploty LM35 - projekt 34
// LM35 má 10 mV/°C. 500L zabrání přetečení součinu na UNO.
// Oprava je popsána u projektu v českém návodu.

// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int val;
  int dat;
  // Na UNO načteme hodnotu ADC v rozsahu 0 až 1023.
  val = analogRead(0);
  dat = (500L * val) / 1024;
  Serial.print("Temp:");
  Serial.print(dat);
  Serial.println("C");
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(500);
}
```

Očekávaný výsledek

Monitor na 9600 baud vypisuje „Temp:“ a teplotu v celých stupních Celsia každých 500 ms. Zahřátí snímače prsty by mělo způsobit pomalý růst hodnoty.

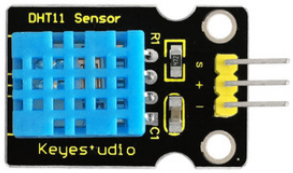


Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Porovnejte s běžným teploměrem a nechte oba přístroje ustálit. Původní celočíselný výpočet zaokrouhluje dolů; jeden krok převodníku při 5 V odpovídá přibližně 0,49 °C.

Projekt 35: Teplota a vlhkost s DHT11

Popis



DHT11 spojuje snímání teploty a relativní vlhkosti s vlastní digitální komunikací. Kalibrační údaje jsou uloženy uvnitř senzoru. Program podle předlohy používá knihovnu dht11.h, vypisuje verzi knihovny, stav komunikace, vlhkost a teplotu. Tuto strukturu ponecháváme.

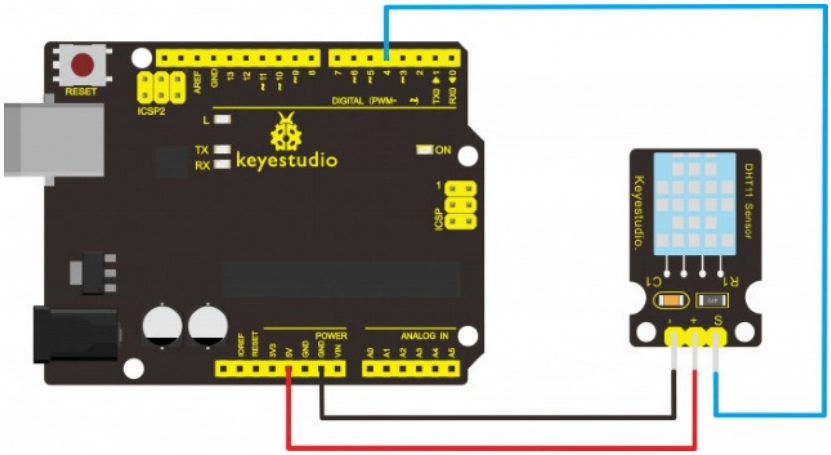
Parametry

- Napájení modulu: 5 V
- Teplota: 0 až 50 °C, uváděná chyba ±2 °C
- Vlhkost: 20 až 90 % RH, uváděná chyba ±5 % RH
- Rozhraní: digitální jednovodičové
- Rozměry: 30 × 20 mm; hmotnost 4 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 3 propojovací vodiče.

Modul	Arduino / spoj
S	D4
+ / V	5 V
- / G	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Oprava HWKITCHEN: naměřené hodnoty se vypisují jen při DHTLIB_OK, aby se při chybě nevydávala stará data za nové měření. Převod na float zajišťuje, že parametr 1 znamená jedno desetinné místo i u knihovny s celočíselnými členy. Hlavička dht11.h a původní rozhraní DHT.read() zůstávají. Mezi čteními nyní vyčkáme 2 s, aby měl pomalý snímač dost času na další měření.

Jak pokus funguje

Knihovna vrátí stav čtení a data. Kontrolní součet pomáhá rozpoznat poškozenou zprávu. Relativní vlhkost RH vyjadřuje nasycení vzduchu vodní párou při dané teplotě.

První ověření a častá chyba

Nejprve očekávejte stav OK v pokojových podmínkách. Desetinné místo ve výpisu samo nezvyšuje rozlišení ani přesnost senzoru. DHT11 neponořujte do vody a mezi odečty ponechte alespoň dvě sekundy jako v ukázce.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 35 - ukázkový program

Projekt_35.ino

```
// Teplota a vlhkost s DHT11 - projekt 35
// Použijte původní dht11.h s DHT.read(), nikoli Adafruit DHT.h.
// Při chybě komunikace se teplota a vlhkost nevypisují.
// Oprava je popsána u projektu v českém návodu.

#include <dht11.h>
// Objekt původní knihovny pro DHT11.
dht11 DHT;
#define DHT11_PIN 4
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
  Serial.println("DHT TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT11LIB_VERSION);
  Serial.println();
  Serial.println("Type,\tstatus,\tHumidity (%),\tTemperature (C)");
}
// loop() Arduino stále znovu opakuje.
void loop() {
  int chk;
  Serial.print("DHT11, \t");
  // Výsledek čtení obsahuje stavový kód.
  chk = DHT.read(DHT11_PIN);
  // Rozlišíme úspěch, chybu kontrolního součtu a timeout.
  switch (chk) {
    case DHTLIB_OK: Serial.print("OK,\t");
    break;
    case DHTLIB_ERROR_CHECKSUM: Serial.print("Checksum error,\t");
    break;
    case DHTLIB_ERROR_TIMEOUT: Serial.print("Time out error,\t");
    break;
    default: Serial.print("Unknown error,\t");
    break;
  }
  // Naměřené hodnoty použijeme jen při úspěšném čtení.
  if (chk == DHTLIB_OK) {
    Serial.print((float)DHT.humidity, 1);
    Serial.print(",\t");
    Serial.println((float)DHT.temperature, 1);
  }
  else {
    Serial.println();
  }
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(2000);
}
```

Očekávaný výsledek

Při 9600 baud se nejprve objeví hlavička a verze knihovny. Další řádky obsahují stav čtení. U platného měření následují vlhkost v % RH a teplota v °C. Při chybě nejprve zkontrolujte zapojení a použitou knihovnu.

Tip HWKITCHEN - vyzkoušejte navíc
Porovnejte běžné prostředí a krátké přiblížení ruky k senzoru. DHT11 nenamáčejte a nefoukejte na něj horkou páru. Vlhkost vzduchu z tohoto modulu není totéž co vlhkost půdy z projektu 24.

Projekt 36: Ultrazvukový dálkoměr

Popis



Senzor vyšle ultrazvukovou dávku a změří čas, za který se ozvěna vrátí. Program vytvoří na TRIG impuls dlouhý 10 µs. Modul potom vyšle osm period přibližně 40 kHz a na ECHO vytvoří impuls, jehož délka odpovídá době letu. Vzdálenost je rychlost zvuku násobená dobou a dělená dvěma, protože zvuk urazí cestu tam i zpět. Předloha používá zjednodušený převod mikrosekundy / 58 na centimetry.

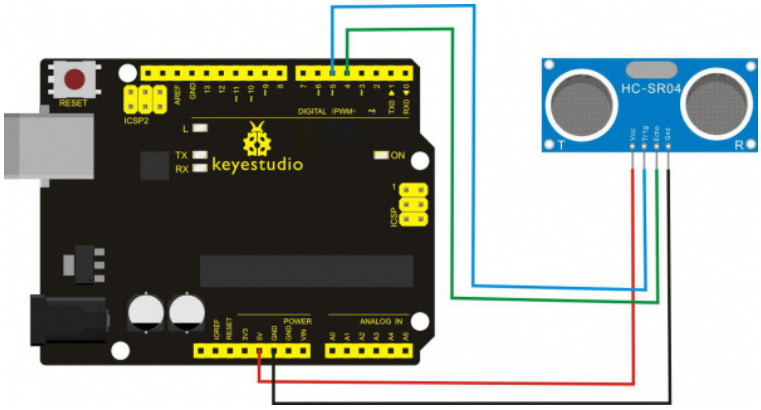
Parametry

- Napájení: 5 V DC
- Proud uváděný předlohou: 15 mA
- Rozsah dle předlohy: 2 až 450 cm
- Detekční úhel: přibližně 15°
- Spouštěcí impuls: alespoň 10 µs
- Výstup ECHO: délka digitálního impulsu
- Přesnost závisí na cíli a prostředí; předloha uvádí až 3 mm

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 4 propojovací vodiče.

Modul	Arduino / spoj
ECHO	D4
TRIG	D5
VCC	5 V
GND	GND



Původní obrázek zapojení: Keyestudio.

Oprava nebo upřesnění předlohy
Opravy HWKITCHEN: napájení je 5 V, nikoli chybné 0,5 V z předlohy. pulseIn() používá unsigned long a časový limit 30 ms. Při chybějícím impulsu vypíšeme zprávu místo zavádějící nuly. Prodleva mezi měřeními je 60 ms; ECHO na D4, TRIG na D5 a převod /58 jsou zachovány.

Jak pokus funguje

ECHO vyjadřuje dobu letu tam a zpět. Dělení 58 je přibližný převod pro běžnou pokojovou teplotu. Nulová doba od pulseIn() znamená chybějící dokončený impuls v limitu, nikoli vzdálenost nula.

První ověření a častá chyba

Použijte rovnou pevnou plochu alespoň několik centimetrů od snímače. Ověřte výsledek pravítkem. Nová ukázka chybějící odraz výslovně hlásí a mezi měřeními vyčká nejméně 60 ms.

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 36 - ukázkový program

Projekt_36.ino

```
// Ultrazvukový dálkoměr - projekt 36
// ECHO=D4, TRIG=D5. Délku impulsu v µs dělíme 58 pro cm.
// Oprava je popsána u projektu v českém návodu.

int inputPin = 4;
int outputPin = 5;
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Stejných 9600 baud nastavte v Sériovém monitoru.
  Serial.begin(9600);
  // Senzorový pin bude vstupem.
  pinMode(inputPin, INPUT);
  // Řídicí pin nastavíme jako výstup.
  pinMode(outputPin, OUTPUT);
  digitalWrite(outputPin, LOW);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  digitalWrite(outputPin, LOW);
  delayMicroseconds(2);
  digitalWrite(outputPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(outputPin, LOW);
  // Čekáme nejvýše 30000 µs = 30 ms na impuls.
  unsigned long doba = pulseIn(inputPin, HIGH, 30000UL);
  // Nula znamená, že měření neposkytlo platný impuls.
  if (doba == 0) {
    Serial.println("Bez odrazu / mimo rozsah");
  }
  else {
    // Převod doby letu tam i zpět na celé centimetry.
    unsigned long vzdalenost = doba / 58;
    Serial.println(vzdalenost);
  }
  // Prodleva je v milisekundách: 1000 ms = 1 s.
  delay(60);
}
```

Očekávaný výsledek

Při 9600 baud se vypisují celé centimetry. Přibližujte a oddalujte rovnou pevnou plochu a porovnejte údaj s pravítkem. Pokud se během limitu nevrátí dokončený impuls, program vypíše „Bez odrazu / mimo rozsah“. Šikmý, měkký nebo malý cíl může být pro snímač nečitelný; zpráva může také znamenat chybné zapojení.

Tip HWKITCHEN - vyzkoušejte navíc
Změřte několik známých vzdáleností a zapište odchylku od pravítka. Potom mírně nakloňte cílovou desku a porovnejte výsledek. Vysvětlete, proč zpráva o chybějícím odrazu neříká, zda je problém v cíli, vodiči nebo napájení.

Projekt 37: Čtyřmístný LED displej

Popis



Modul používá čtyři sedmisegmentové číslice a řadič TM1637. Pro ovládání stačí signály CLK a DIO, napájení a zem. Ukázka předlohy postupně zobrazuje minuty a sekundy pomocí čtyř vnořených smyček. Zachováváme tento postup včetně rozsvícené dvojtečky.

Parametry

- Napájení: 5 V DC
- Displej: čtyři číslice, výška 0,36 palce, společná anoda
- Řadič: TM1637
- Signály: CLK, DIO
- Teplota okolí dle předlohy: -40 až +85 °C
- Montážní otvory: 2× průměr 3 mm
- Rozměry: 45 × 21 × 10 mm; hmotnost 7,2 g

Zapojení

Připravte UNO, datový USB kabel, příslušný modul a 4 propojovací vodiče.

Modul	Arduino / spoj
CLK	D6
DIO	D5
VCC	5 V
GND	GND

Oprava nebo upřesnění předlohy
Upřesnění HWKITCHEN: použijte knihovnu s třídou TM1637 a funkcemi init(), set(), point() a display(), nikoli odlišné rozhraní TM1637Display. Zapojení přepisujeme do přehledné tabulky, protože původní obrázek zapojení obsahuje grafické vady.

Jak pokus funguje

Řadič TM1637 rozsvěcí segmenty podle čísel od Arduina. CLK a DIO tvoří jeho vlastní dvouvodičové rozhraní, nikoli I2C. Vnořené smyčky postupně mění jednotlivé číslice.

První ověření a častá chyba

Po resetu se čas vrátí na začátek. delay(1000) plus čas ostatních příkazů způsobují odchylku; pro dlouhodobě přesné hodiny je vhodný obvod reálného času (RTC).

Příslušný projekt v originální dokumentaci Keyestudio

Projekt 37 - ukázkový program

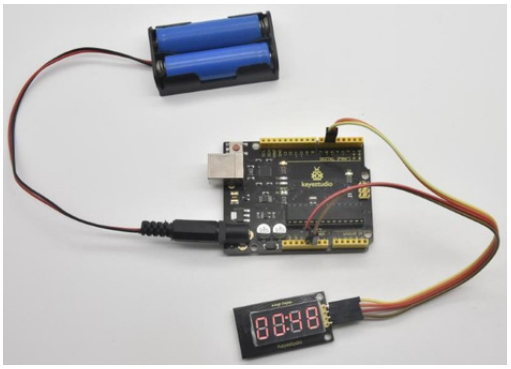
Projekt_37.ino

```
// Čtyřmístný LED displej - projekt 37
// a,b jsou desítky a jednotky minut; c,d patří sekundám.
// Původní smyčky projdou 00:00 až 59:59, pak začnou znovu.

#include "TM1637.h"
#define CLK 6
#define DIO 5
// Objekt displeje s piny CLK a DIO.
TM1637 tm1637(CLK, DIO);
// setup() se provede jednou po zapnutí nebo resetu.
void setup() {
  // Inicializace řadiče displeje.
  tm1637.init();
  // Nastavení obvyklého jasu podle knihovny.
  tm1637.set(BRIGHT_TYPICAL);
}
// loop() Arduino stále znovu opakuje.
void loop() {
  // Rozsvícení dvojtečky mezi minutami a sekundami.
  tm1637.point(POINT_ON);
  // Desítky minut: 0 až 5.
  for (int a = 0; a < 6; a++) {
    tm1637.display(0, a);
    // Jednotky minut: 0 až 9.
    for (int b = 0; b < 10; b++) {
      tm1637.display(1, b);
      // Desítky sekund: 0 až 5.
      for (int c = 0; c < 6; c++) {
        tm1637.display(2, c);
        // Jednotky sekund: 0 až 9.
        for (int d = 0; d < 10; d++) {
          tm1637.display(3, d);
          // Prodleva je v milisekundách: 1000 ms = 1 s.
          delay(1000);
        }
      }
    }
  }
}
```

Očekávaný výsledek

První dvě číslice představují minuty, druhé dvě sekundy. Hodnota postupuje od 00:00 do 59:59 a poté se cyklus opakuje. V programu zůstává delay(1000), takže jde o demonstrační časovač, nikoli o přesné hodiny.



Ilustrace výsledku z předlohy Keyestudio.

Tip HWKITCHEN - vyzkoušejte navíc
Změňte jas v tm1637.set() v povoleném rozsahu 0 až 7. Jako samostatné rozšíření lze časování převést na millis(), aby program během čekání mohl obsluhovat tlačítka.

Přehled oprav a doplnění

Ze 40 ukázek zachovává 29 stejnou posloupnost příkazů jako webová předloha; doplněny jsou české komentáře a přehledná sazba. V 11 ukázkách jsou následující označené opravy. Pořadí ani základní účel žádného projektu se nemění.

Projekt 2

Pojmenování RGB pinů podle obrázku; omezení záporného PWM.

Projekt 5

Doplněna chybějící kladná půlperioda ve druhé sérii pulzů.

Projekt 13

Čekání 2 s přesunuto z přerušení do loop(); přerušení pouze nastaví příznak.

Projekt 19, ukázka 2

Ochrana proti dělení nulou a logaritmu neplatné hodnoty.

Projekt 27

Opraven přibližný převod na desetiny voltu pro nominální referenci 5 V a dělič 5 : 1.

Projekt 29, ukázka 2

Délka dat RC5 změněna z 8 na 12 bitů (toggle + adresa + příkaz); startovní bity přidává staré rozhraní knihovny.

Projekt 29, ukázka 3

Doplněno nastavení vestavěné LED jako výstupu.

Projekt 32

Odstraněno delay z přerušení; volatile a atomické čtení/nulování COUNT; inicializace před přerušením.

Projekt 34

500L vynutí 32bitový výpočet a zabrání přetečení int na AVR.

Projekt 35

Výpis pouze platných dat; explicitní float pro desetinná místa; interval čtení prodloužen na 2 s.

Projekt 36

Doba impulsu je unsigned long; timeout 30 ms se hlásí jako chybějící odraz; mezi pokusy je prodleva 60 ms. Piny a převod /58 zůstávají.

Opravy textu a zapojení

Vysvětleny jsou zejména obrácené nebo zavádějící popisy logických úrovní, záměna Hallova a jazýčkového snímače, dvojí význam NC (nevyužitý vývod oproti klidovému kontaktu relé), nezapojené NC u projektů 26 a 27, chybná zmínka o D11 u vysílače, potřeba druhého UNO pro dvojici komunikačních programů a překlep 0,5 V u ultrazvuku. Rozporné nebo neověřené parametry nejsou vydávány za zaručené vlastnosti celé desky.

Doplnění HWKITCHEN

Každá kapitola obsahuje české komentáře, vysvětlení fungování, první ověření výsledku a častou chybu. Samostatně označený tip rozšiřuje základní pokus. Úvod vydání 3.0 vysvětluje Arduino, analogové a digitální signály, PWM, elektrické veličiny, pinové značení a čtení programu. Doplněn je český instalační postup, orientace v knihovnách, bezpečné zacházení, tabulky pinů, klikací obsah a samostatné soubory pro Arduino IDE. Původní anglické řetězce ve výpisech většinou zůstávají, aby odpovídaly předloze.

Rychlé řešení problémů

Program nelze přeložit

Zkontrolujte celou zkopírovanou ukázku, jedinou funkci setup() a loop(), zvolenou desku a přesnou knihovnu. Chyba „No such file“ znamená chybějící hlavičku; chybějící třída nebo metoda často znamená jinou verzi knihovny.

Program se nepodaří nahrát

Ověřte správný port, datový kabel a výběr desky. Zavřete jiné aplikace používající stejný sériový port. Nejprve vyzkoušejte vestavěný Blink bez senzorů.

Monitor zobrazuje nesmyslné znaky

Rychlost musí odpovídat Serial.begin(): většinou 9600, u projektu 24 je 57600 a u projektu 30 je 115200 baud.

Senzor nereaguje

Odpojte napájení a porovnejte popisky pinů s tabulkou. D3 není A3. Ověřte společnou zem a správný modul. Nastavitelný práh se musí přizpůsobit prostředí.

LED se chová obráceně

LOW může znamenat aktivní snímač. Původní programy nepoužívají jednotnou polaritu všech modulů. Přečtěte očekávaný výsledek konkrétní kapitoly.

Hodnota kolísá

Změřte klidový stav, ověřte napájení, polohu a kontakt se vzorkem. Průměrování může pomoci, ale u krátkého zvuku nebo klepnutí naopak potřebujete rychlejší snímání.

Po kopírování chybí řádky nebo odsazení

Použijte odpovídající .ino ze ZIP. Každá ukázka má vlastní složku a shodně pojmenovaný soubor, takže se otevře přímo v Arduino IDE.

Předloha, zpracování a zdroje

Předloha

Keyestudio: KS0487 Keyestudio 37 in 1 Sensor Kit v3.0. Webová dokumentace a původní obrázky použité v jednotlivých projektech. Původní podklady byly načteny 6. 11. 2026; technická a výuková revize proběhla 9. 11. 2026.

docs.keyestudio.com/projects/KS0487/en/latest/ks0487.html

Zdrojový projekt KS0487 na GitHubu Keyestudio

Doplňující technické podklady

Arduino: UNO R3, rozhraní a technické údaje
Arduino: PWM a jeho odlišení od analogového napětí
Arduino: attachInterrupt() a omezení obsluhy přerušení
Arduino: volatile a sdílení proměnných
Arduino-IRremote: knihovna a rozdíly verzí
Seeed Studio: původní rozhraní TM1637

Vydání a odpovědnost za zpracování

České zpracování a doplnění: HWKITCHEN. Vydání 3.0, listopad 2026. Předloha, původní fotografie a obrázky zapojení: Keyestudio. Logo a obchodní označení HWKITCHEN patří jejich vlastníkovi; označení Keyestudio a Arduino se používají k identifikaci výrobků a platformy.

Ukázky slouží k výuce na UNO R3. Fyzický test všech modulů ani úplný překlad všech programů s původními knihovnami nebyl proveden. Před vlastním rozšiřováním ověřte zapojení, verzi součástek a chování základního pokusu.

O původu tohoto návodu
Tento český návod byl volně zpracován podle předlohy „KS0487 Keyestudio 37 in 1 Sensor Kit v3.0“ od Keyestudio a doplněn HWKITCHEN o české komentáře programů, vysvětlení, upozornění na chyby, bezpečnostní poznámky a náměty na další pokusy. Zachovává pořadí a základní obsah projektů předlohy. Věcné opravy a vlastní doplnění jsou v textu označeny.

www.hwkitchen.cz