



ZESTAW STARTOWY OTRONIC ARDUINO UNO

Zestaw startowy Arduino UNO: tutorial i 16 projektów

Krok po kroku: czym jest Arduino, co jest w twoim zestawie, jak działa Codey i 16 projektów do samodzielnego zbudowania.

codey.online/pl/starter-kits/uno/tutorial



Spis treści

Witaj! Dasz radę to zrobić! 🚀	3
Co to jest Arduino UNO?	4
Co jest w twoim zestawie startowym?	5
Breadboard	6
Kabelki jumper	7
LED (lampka)	8
Rezystor 220 Ω	9
Przycisk	10
Aktywny buzzer	11
Czujnik światła (LDR)	12
Czujnik temperatury DHT11	13
Czujnik odległości HC-SR04	14
Czujnik ruchu (PIR)	15
Ekranik OLED 0,91"	16
Serwomotor SG90	17
Pierścień LED RGB (8 diod LED)	18
Przełącznik 5V	19
Klips do baterii 9V	20
Co to jest vibe coding?	21
Utwórz darmowe konto	22
Tak działa Codey	23
Podłączanie i wgrywanie	24
Do dzieła! 🎉	25
Twój pierwszy układ na breadboardzie	26
Migacz	27
Sygnalizacja świetlna	28
Przycisk światła	29
Dzwonek do drzwi	30
Tęczowy pierścień	31
Mini-stacja pogody	32
Czujnik parkowania	33
Lampka nocna	34
Alarm ruchu	35
Szlaban	36
Gra na refleks	37
Elektroniczna kostka	38
Czujnik parkowania deluxe	39
Miernik odległości z paskiem świetlnym	40
Sprytny wentylator	41
Automatyczny szlaban	42

ZESTAW STARTOWY OTRONIC ARDUINO UNO

Witaj! Dasz radę to zrobić! 🚀

Z tym zestawem startowym zbudujesz własne sprytne urządzenia. Stację pogody, czujnik parkowania, alarm... A najlepsze jest to: nie musisz umieć programować. Codey ci pomoże.

Przejdź przez ten krótki tutorial. Wtedy dowiesz się, co jest w twoim zestawie i co możesz z tym zrobić. Potem od razu zaczniesz swój pierwszy projekt.

Tak to działa:

- 2 Czym jest Arduino i co możesz z nim zrobić?
- 3 Wszystkie części z twojego zestawu, jedna po drugiej
- 4 Czym jest vibe coding?
Programowanie z AI
- 5 Tworzenie darmowego konta
- 6 Tak działa Codey
- 7 Podłączanie Arduino i wgrywanie kodu
- 8 Startujemy z 16 projektami

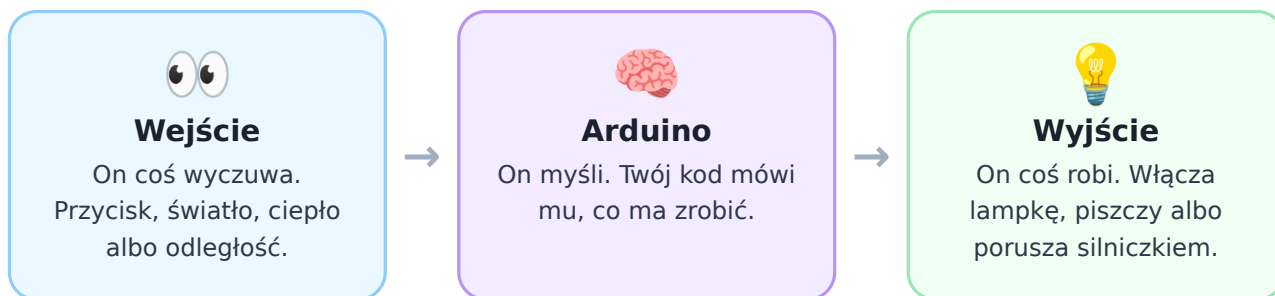


🕒 To zajmie około 15 minut. Zawsze możesz coś pominąć i wrócić później.

Co to jest Arduino UNO?

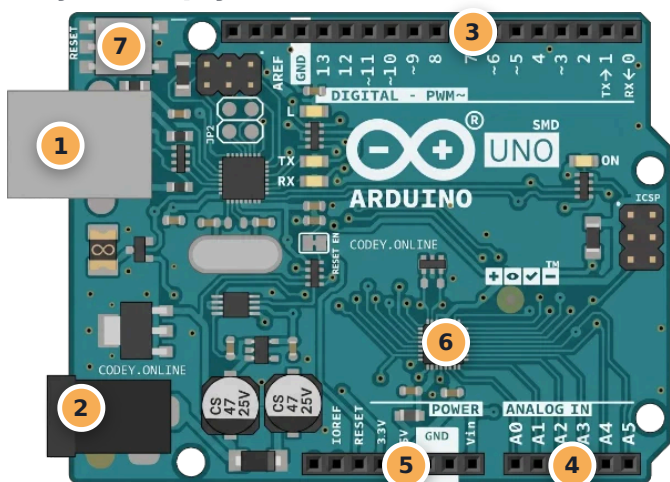
Arduino to mały komputer. Mówisz mu, co ma zrobić, a on to robi. Wciąż od nowa, bez zmęczenia.

Tak myśli Arduino



Przykład: czujnik odległości widzi, że podchodzisz blisko (wejście). Arduino myśli: "za blisko!" I brzęczyk zaczyna piszczeć (wyjście). Tak działa czujnik parkowania w samochodzie.

Co jest na płytce?



- 1 Port USB: tutaj podłączasz kabel do komputera. Tędy Arduino dostaje prąd i kod.
- 2 Gniazdo zasilania: na baterię albo zasilacz (7-12V), gdy nie używasz komputera.
- 3 Piny cyfrowe od D0 do D13: włączone albo wyłączone. Do lampek, przycisków i brzęczyków.
- 4 Piny analogowe od A0 do A5: mierzą ile czegoś jest. Na przykład ile jest światła.
- 5 Piny zasilania: 5V (plus) i GND (minus) dają prąd twoim częściom.
- 6 Chip: mózg. Tutaj trafia twój kod.
- 7 Przycisk reset: uruchamia twój kod od nowa.

Dwa słowa, które będziesz często widzieć:

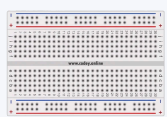
- **5V** to plus. Stąd płynie prąd (5 volt, tak jak z ładowarki do telefonu).
- **GND** (ground) to minus. Tędy prąd wraca z powrotem. Każda część potrzebuje przewodu do GND.

Piny to małe otworki z boku. Wkładasz w nie przewody. Pin może być **wejściem** (słucha) albo **wyjściem** (daje prąd). Ustawiasz to w kodzie.

Co jest w twoim zestawie startowym?

Twój zestaw jest pełen części. Niektóre coś czują (czujniki), a inne coś robią (lampki, buzzer, silniczek). Kliknij Dalej, a poznasz je po kolei.

Podstawa: z tym zbudujesz wszystko



Breadboard

Płytko do budowania:
budujesz bez
lutowania

W twoim zestawie: 1x



Kabelki jumper

Kabelki Dupont:
przewody, które łączą
wszystko

W twoim zestawie: 40x



Czerwony, czerwony, brązowy = 220 Ω

Rezystor 220 Ω

Hamuje prąd

W twoim zestawie: 10x



Klips do baterii 9V

Niech twój projekt
działa bez komputera

W twoim zestawie: 1x

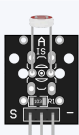
Input: te części coś czują albo mierzą



Przycisk

Mały przycisk do
naciskania

W twoim zestawie: 5x



Czujnik światła (LDR)

Mierzy, jak jasno albo
ciemno jest

W twoim zestawie: 1x



Czujnik temperatury DHT11

Mierzy temperaturę i
wilgotność powietrza

W twoim zestawie: 1x



Czujnik odległości HC-SR04

Mierzy odległość
dźwiękiem, tak jak
nieoperz

W twoim zestawie: 1x



Czujnik ruchu (PIR)

Widzą, czy ktoś się
rusza

W twoim zestawie: 1x

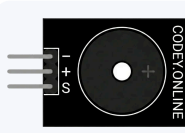
Output: te części coś robią



LED (lampka)

Mała lampka w 5
kolorach

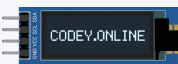
W twoim zestawie: 10x



Aktywny buzzer

Robi piknięcia

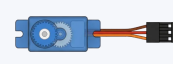
W twoim zestawie: 1x



Ekranik OLED 0,91"

Mały ekran do tekstu i
liczb

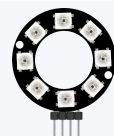
W twoim zestawie: 1x



Serwomotor SG90

Mały silnik, który
obraca się do
wybranej pozycji

W twoim zestawie: 1x



Pierścień LED RGB (8 diod LED)

8 sprytnych lampek w
każdym kolorze

W twoim zestawie: 1x



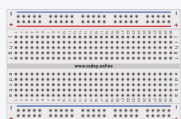
Przełącznik 5V

Przełącznik, który
może przełączać
twoje Arduino

W twoim zestawie: 1x



Część 1 z 15



Breadboard

Płytko do budowania: budujesz bez lutowania

W twoim zestawie: 1x

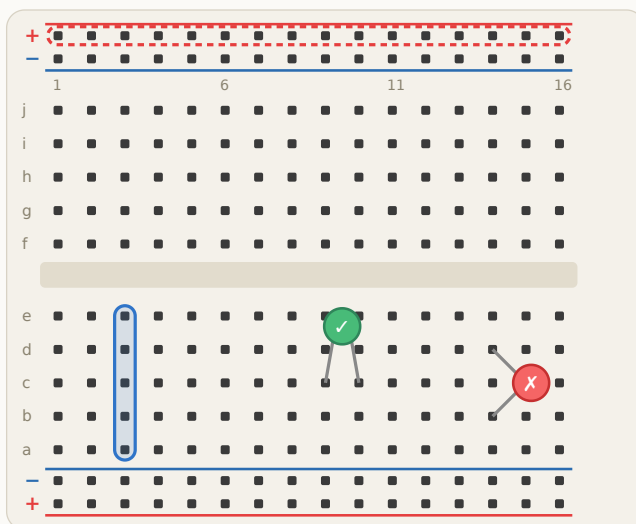
Co to jest?

Breadboard to mała płytko pełna dziurek. Wkładasz w nią części i kabelki. Pod dziurkami są metalowe paski, które łączą niektóre dziurki ze sobą. Tak robisz układ bez lutowania. Coś poszło źle? Po prostu wyjmij i spróbuj od nowa.

Tak to podłączysz

Tak to działa:

- **Rzędy po 5:** w środkowej części zawsze 5 dziurek obok siebie jest połączonych (a-e i f-j). Wszystko, co włożysz do tego samego rzędu, jest ze sobą połączony.
- **Przerwa pośrodku** rozdziela dwie strony. Lewa strona (a-e) i prawa strona (f-j) *nie* są więc połączone.
- **Szyny zasilania (+ i -)** przy długich bokach są połączone na całej długości. Połącz + z 5V, a - z GND w Arduino. Teraz masz zasilanie wszędzie.



- Szyna zasilania: cała długość jest połączona
- 5 dziurek w rzędzie jest połączonych
- X Źle: obie nóżki są w tym samym rzędzie

- Przerwa: lewa i prawa strona nie są połączone
- ✓ Dobrze: nóżki są w różnych rzędach

Co możesz z tym zrobić?

- Testowanie układów bez lutowania
- Dawanie zasilania kilku częściom naraz przez szyny + i -
- Ładne połączenie LED z rezystorem w szeregu

Wskazówka

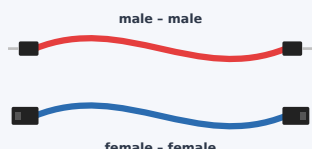
Nigdy nie wkładaj dwóch nóżek jednej części do tego samego rzędu z 5 dziurkami. Wtedy są ze sobą połączone i prąd po prostu je omija. Lepiej włoż część przez przerwę pośrodku.

🔧 Wypróbuj to od razu! W ćwiczeniu „Twój pierwszy układ na breadboardzie” zbudujesz LED z rezystorem na breadboardzie i sprawisz, że będzie świecić i migać. →

Projekty z tą częścią

★ Twój pierwszy układ na breadboardzie

Część 2 z 15



Kabelki jumper

Kabelki Dupont: przewody, które łączą wszystko

W twoim zestawie: 40x

Co to jest?

Kabelkami jumper łączysz Arduino z częściami. Nazywa się je też **kabelkami Dupont**, od czarnych wtyczek na końcach. W twoim zestawie są dwa rodzaje:

- **Male-male** (pin po obu stronach): do wkładania w Arduino i breadboard.
- **Female-female** (dziurka po obu stronach): do wsuwania na piny modułu.

Tak to podłączysz

Włóż kabel mocno w dziurkę albo nasuń go na pin. Jeśli połączysz kabel male i kabel female, masz kabel male-female: przydaje się, gdy chcesz podłączyć moduł prosto do Arduino.

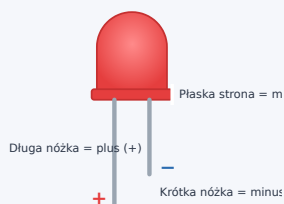
Co możesz z tym zrobić?

- Łączenie Arduino z breadboard
- Podłączanie modułów z czujnikami
- Doprowadzanie zasilania (5V) i GND do części

Wskazówka

Używaj stałych kolorów: czerwony dla 5V i czarny dla GND. Wtedy od razu widzisz, czy zasilanie jest dobrze podłączone.

Część 3 z 15



LED (lampka)

Mała lampka w 5 kolorach

W twoim zestawie: 10x

Co to jest?

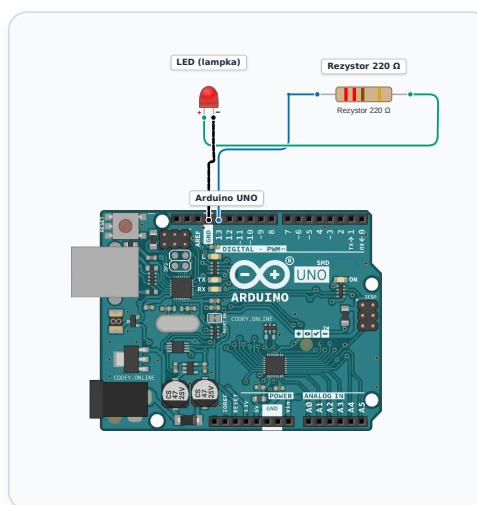
LED to mała lampka, która zużywa bardzo mało prądu. W twoim zestawie jest 10 LEDs w 5 kolorach. LED działa tylko w jedną stronę, więc musisz włożyć ją dobrze.

Tak to podłączysz

- **Długa nóżka** to plus (+). Idzie przez rezystor 220 Ω do pinu Arduino.
- **Krótka nóżka** to minus (-). Idzie do GND. Po tej stronie brzeg lampki jest też płaski.

Zawsze używaj rezystora! Bez rezystora LED dostanie za dużo prądu i się zepsuje.

Schemat podłączenia



- Arduino UNO · D13 → Rezystor 220 Ω · 1
- Rezystor 220 Ω · 2 → LED (lampka) · ANODE
- LED (lampka) · CATHODE → Arduino UNO · GND

Co możesz z tym zrobić?

- Migające światełko
- Sygnalizacja świetlna
- Lampka, która pokazuje, czy coś jest włączone czy wyłączone

Wskazówka

Twoja LED nic nie robi? Odwróć ją. Działa tylko w jedną stronę.

Projekty z tą częścią

★ Twój pierwszy układ na breadboardzie

★ Migacz

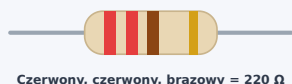
★ Sygnalizacja świetlna

★ Przycisk światła

★★ Alarm ruchu

★★ Gra na refleks

Część 4 z 15

Czerwony, czerwony, brązowy = 220 Ω

Rezystor 220 Ω

Hamuje prąd

W twoim zestawie: 10x

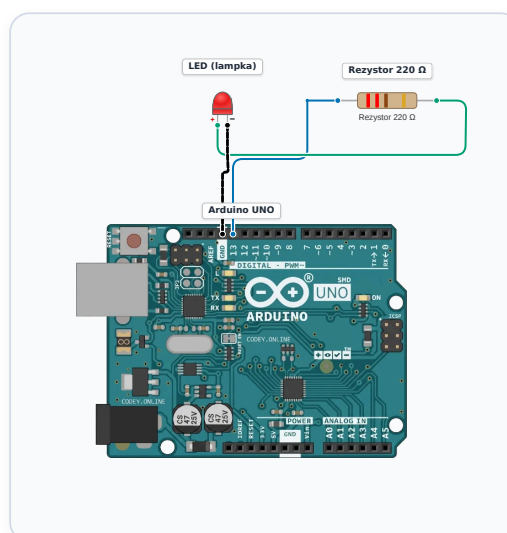
Co to jest?

Rezystor trochę hamuje prąd. To jak kran, który odkręcasz tylko do połowy. Dzięki temu LED dostaje dokładnie tyle prądu, ile trzeba, i się nie psuje. Rezystory w twoim zestawie mają 220 ohm (Ω). Poznasz to po kolorowych paskach: czerwony, czerwony, brązowy.

Tak to podłączysz

Rezystor nie ma plusa ani minusa: możesz włożyć go w dowolną stronę. Włóż go między pin Arduino a długą nóżkę LED. Jedna strona idzie do pinu, druga strona do LED.

Schemat podłączenia



- Arduino UNO · D13 → Rezystor 220 Ω · 1
- Rezystor 220 Ω · 2 → LED (lampka) · ANODE
- LED (lampka) · CATHODE → Arduino UNO · GND

Co możesz z tym zrobić?

- Ochrona każdej LED
- Ograniczanie prądu do części

Wskazówka

Jeden rezystor na jedną LED. Używasz trzech LEDs? Wtedy potrzebujesz też trzech rezystorów.

Projekty z tą częścią

★ Twój pierwszy układ na breadboardzie

★ Migacz

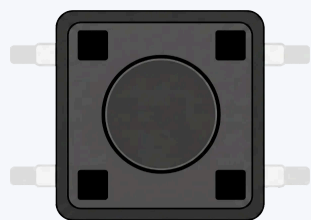
★ Sygnalizacja świetlna

★ Przycisk światła

★★ Alarm ruchu

★★★ Gra na refleks

Część 5 z 15



Przycisk

Mały przycisk do naciskania

W twoim zestawie: 5x

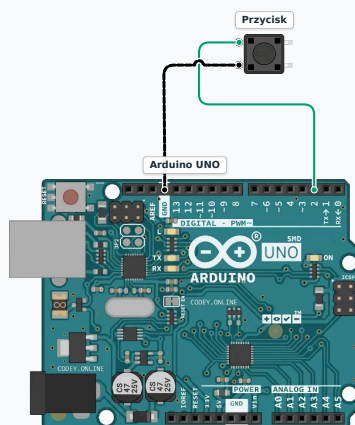
Co to jest?

Przycisk łączy styki tak długo, jak go naciskasz. Puszczasz go i połączenie znika. Twoje Arduino może wyczuć, czy przycisk jest wciśnięty. W zestawie masz 5 przycisków.

Tak to podłączysz

Przycisk ma 4 nóżki. Potrzebujesz tylko 2: jedną do pinu (na przykład D2) i jedną do GND. W kodzie ustawiasz pin na `INPUT_PULLUP`. Wtedy pin ma HIGH, gdy nie naciskasz, i LOW, gdy naciskasz.

Schemat podłączenia



● Arduino UNO · D2 → Przycisk · 1

● Arduino UNO · GND → Przycisk · 2

Co możesz z tym zrobić?

- Włączać i wyłączać lampkę
- Dzwonek do drzwi
- Gra: kto naciśnie najszybciej?

Wskazówka

Nie wiesz, które nóżki wybrać? Weź dwie nóżki po przekątnej. Wtedy zawsze będzie dobrze. Włóż przycisk nad rowkiem w breadboard.

Projekty z tą częścią

★ Przycisk światła

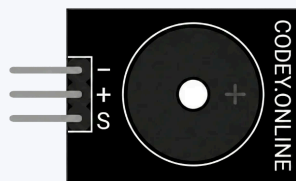
★ Dzwonek do drzwi

★★ Szlaban

★★ Gra na refleks

★★ Elektroniczna kostka

Część 6 z 15



Aktywny buzzer

Robi piknięcia

W twoim zestawie: 1x

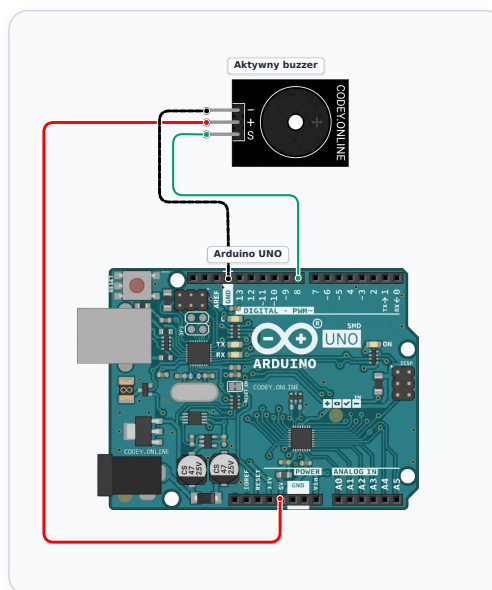
Co to jest?

Buzzer wydaje dźwięk „pik”. To **aktywny** buzzer: pika sam, gdy tylko dostanie prąd. Musisz go tylko włączać i wyłączać, tak jak lampkę.

Tak to podłączysz

- S (sygnał) do pinu, na przykład D8
- + do 5V
- - do GND

Schemat podłączenia



● Arduino UNO · GND → Aktywny buzzer · -

● Arduino UNO · 5V → Aktywny buzzer · +

● Arduino UNO · D8 → Aktywny buzzer · S

Co możesz z tym zrobić?

- Alarm
- Czujnik parkowania, który pika szybciej, gdy jesteś bliżej
- Dzwonek do drzwi albo timer

Wskazówka

Chcesz krótkie piknięcie? Włącz buzzer, poczekaj 100 milisekonden i wyłącz go znowu.

Projekty z tą częścią

★ Dzwonek do drzwi

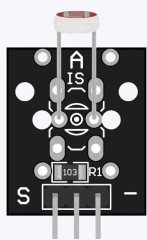
★★ Czujnik parkowania

★★ Alarm ruchu

★★★★ Czujnik parkowania deluxe

★★★★ Automatyczny szlaban

Część 7 z 15



Czujnik światła (LDR)

Mierzy, jak jasno albo ciemno jest

W twoim zestawie: 1x

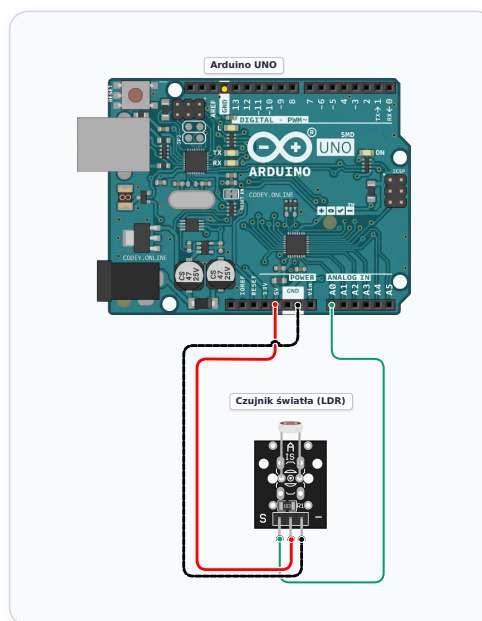
Co to jest?

Ten czujnik mierzy, ile jest światła. Twoje Arduino odczytuje liczbę od 0 do 1023. Przy dużej ilości światła dostaniesz inną liczbę niż w ciemności. Tak Arduino wie, czy jest dzień, czy noc.

Tak to podłączysz

- S do pinu analogowego, na przykład A0
- + do 5V
- - do GND

Schemat podłączenia



● Arduino UNO · A0 → Czujnik światła (LDR) · S

● Arduino UNO · 5V → Czujnik światła (LDR) · +

● Arduino UNO · GND → Czujnik światła (LDR) · -

Co możesz z tym zrobić?

- Lampka nocna, która sama się włącza
- Miernik światła
- Alarm, który włącza się, gdy ktoś otworzy szufladę

Wskazówka

Zobacz w monitorze szeregowym, jaką liczbę widzisz. Potem przytrzymaj rękę nad czujnikiem. Tak zobaczysz, jaka liczba oznacza "ciemno".

Projekty z tą częścią

★★ Lampka nocna

Część 8 z 15



Czujnik temperatury DHT11

Mierzy temperaturę i wilgotność powietrza

W twoim zestawie: 1x

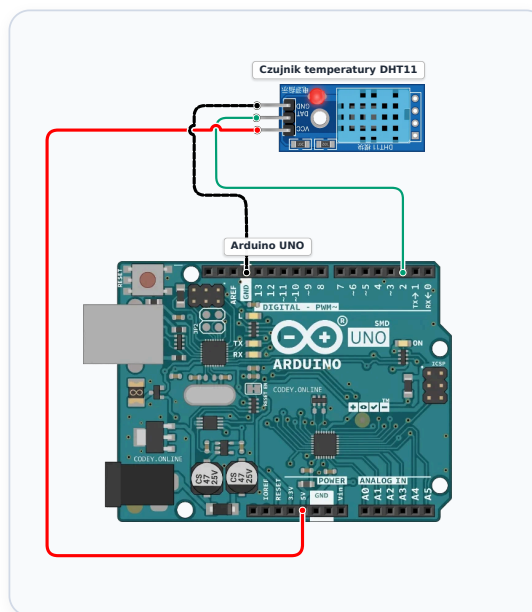
Co to jest?

DHT11 mierzy, jak ciepło jest (w stopniach Celsius) i jak wilgotne jest powietrze (w procentach). Co 2 sekundy daje nowy pomiar.

Tak to podłączysz

- **DAT** (dane) do pinu, na przykład D2
- **VCC** do 5V
- **GND** do GND

Schemat podłączenia



- Arduino UNO · 5V → Czujnik temperatury DHT11 · VCC
- Arduino UNO · GND → Czujnik temperatury DHT11 · GND
- Arduino UNO · D2 → Czujnik temperatury DHT11 · DAT

Co możesz z tym zrobić?

- Mini stacja pogodowa
- Wentylator, który włącza się, gdy jest za ciepło
- Ostrzeżenie, gdy w łazience jest za wilgotno

Wskazówka

Dmuchnij na czujnik. Od razu zobaczysz, jak wilgotność powietrza rośnie!

Projekty z tą częścią

★★★ Mini-stacja pogody

★★★★ Sprytny wentylator

Część 9 z 15



Czujnik odległości HC-SR04

Mierzy odległość dźwiękiem, tak jak nietoperz

W twoim zestawie: 1x

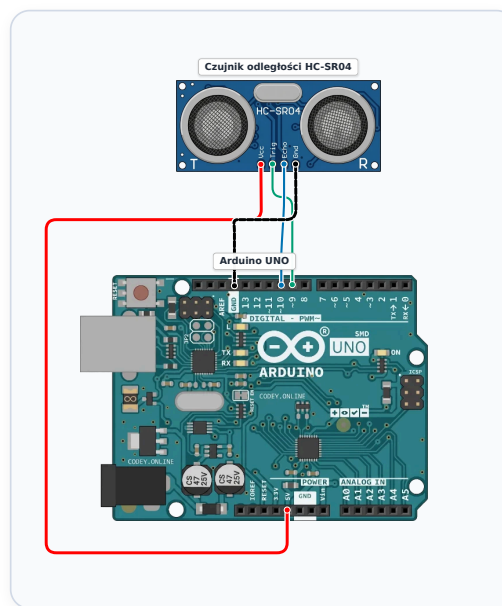
Co to jest?

Ten czujnik wysyła dźwięk, którego nie słyszysz. Dźwięk odbija się od czegoś i wraca. Czujnik mierzy, jak długo to trwa. Dzięki temu Arduino wie, jak daleko coś jest: od 2 cm do około 4 meter.

Tak to podłączysz

- **Trig** (wysyłanie) do D9
- **Echo** (odbieranie) do D10
- **Vcc** do 5V
- **Gnd** do GND

Schemat podłączenia



- Arduino UNO · 5V → Czujnik odległości HC-SR04 · Vcc
- Arduino UNO · GND → Czujnik odległości HC-SR04 · Gnd
- Arduino UNO · D9 → Czujnik odległości HC-SR04 · Trig
- Arduino UNO · D10 → Czujnik odległości HC-SR04 · Echo

Co możesz z tym zrobić?

- Czujnik parkowania
- Miernik odległości
- Szlaban, który sam się otwiera

Wskazówka

Te dwoje „oczu” to głośnik i mikrofon. Skieruj je prosto na to, co chcesz zmierzyć.

Projekty z tą częścią

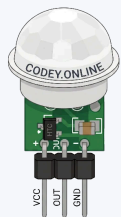
★★ Czujnik parkowania

★★★★ Czujnik parkowania deluxe

★★★★ Miernik odległości z paskiem świetlnym

★★★★ Automatyczny szlaban

Część 10 z 15



Czujnik ruchu (PIR)

Widzą, czy ktoś się rusza

W twoim zestawie: 1x

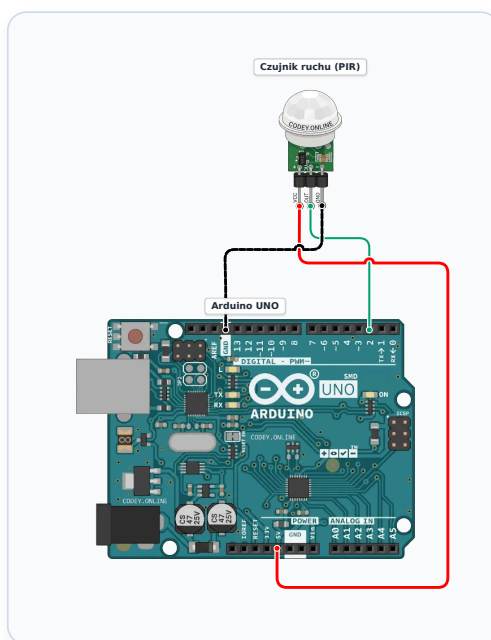
🔍 Co to jest?

Czujnik PIR wyczuwa ciepło ludzi i zwierząt. Czy coś ciepłego porusza się przed czujnikiem? Wtedy jego wyjście staje się HIGH. Tak działają też lampy, które same się włączają, gdy przechodzisz obok.

🔌 Tak to podłączysz

- **OUT** do pinu, na przykład D2
- **VCC** do 5V
- **GND** do GND

Schemat podłączenia



- Arduino UNO · 5V → Czujnik ruchu (PIR) · VCC
- Arduino UNO · GND → Czujnik ruchu (PIR) · GND
- Arduino UNO · D2 → Czujnik ruchu (PIR) · OUT

✨ Co możesz z tym zrobić?

- Alarm antywłamaniowy
- Lampa, która włącza się, gdy wchodzi się
- Liczenie, ile razy ktoś przechodzi obok

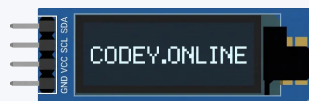
💡 Wskazówka

Po uruchomieniu czujnik potrzebuje kilku sekund, żeby przyzwycząć się do pokoju. Stań wtedy chwilę spokojnie.

Projekty z tą częścią

★★ Alarm ruchu

Część 11 z 15



Ekranik OLED 0,91"

Mały ekran do tekstu i liczb

W twoim zestawie: 1x

Co to jest?

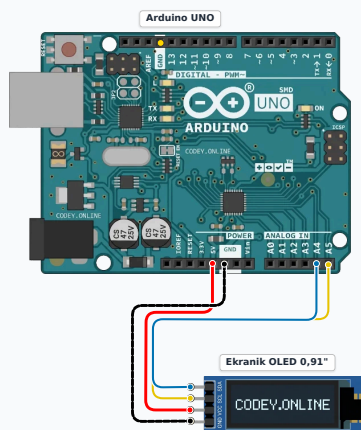
Mały, wyraźny ekranik 128 na 32 pixels. Możesz pokazać na nim tekst, liczby i proste rysunki. Na przykład temperaturę albo odległość.

Tak to podłączysz

Ekran rozmawia przez **I2C**: tylko dwa przewody danych.

- **SDA** do A4
- **SCL** do A5
- **VCC** do 5V
- **GND** do GND

Schemat podłączenia



- Arduino UNO · A4 → Ekranik OLED 0,91" · SDA
- Arduino UNO · A5 → Ekranik OLED 0,91" · SCL
- Arduino UNO · 5V → Ekranik OLED 0,91" · VCC
- Arduino UNO · GND → Ekranik OLED 0,91" · GND

Co możesz z tym zrobić?

- Pokazywanie temperatury
- Pokazywanie odległości w centymetrach
- Pokazywanie wyniku albo czasu reakcji

Wskazówka

Ekran nadal jest czarny? Sprawdź, czy SDA i SCL nie są zamienione miejscami.

Projekty z tą częścią

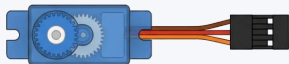
★★★ Mini-stacja pogody

★★★ Gra na refleks

★★★★ Czujnik parkowania deluxe

★★★★ Sprytny wentylator

Część 12 z 15



Serwomotor SG90

Mały silnik, który obraca się do wybranej pozycji

W twoim zestawie: 1x

Co to jest?

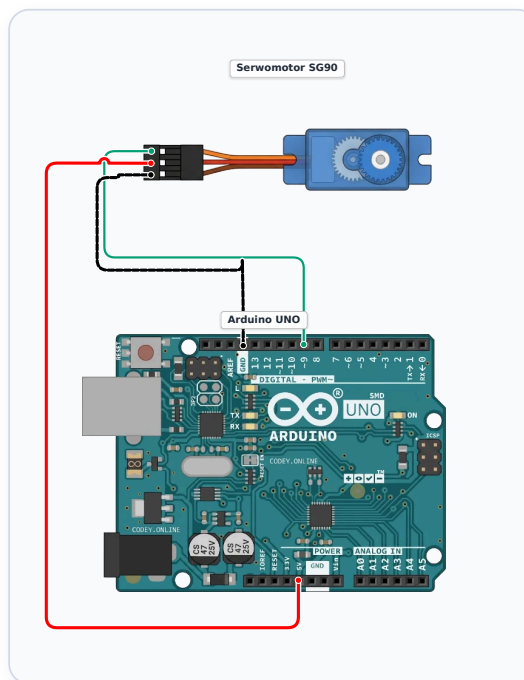
Servo to mały silnik, który możesz obrócić do dokładnej pozycji, od 0 do 180 graden. Mówisz na przykład „idź na 90 graden”, a on obraca się dokładnie tam. Na ośkę zakładasz ramię.

Tak to podłączysz

Servo ma trzy przewody:

- **Pomarańczowy albo żółty** (sygnał) do pinu, na przykład D9
- **Czerwony** do 5V
- **Brązowy** do GND

Schemat podłączenia



● Arduino UNO · GND → Serwomotor SG90 · GND

● Arduino UNO · 5V → Serwomotor SG90 · VCC

● Arduino UNO · D9 → Serwomotor SG90 · PWM

Co możesz z tym zrobić?

- Szlaban
- Wskazówka, która pokazuje temperaturę
- Machające ramię robota

Wskazówka

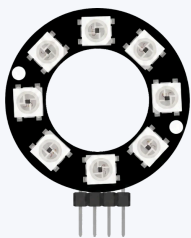
Nie obracaj ramienia ręką. Wtedy zębatki w środku mogą się zepsuć.

Projekty z tą częścią

★★ Szlaban

★★★★ Automatyczny szlaban

Część 13 z 15



Pierścień LED RGB (8 diod LED)

8 sprytnych lampek w każdym kolorze

W twoim zestawie: 1x

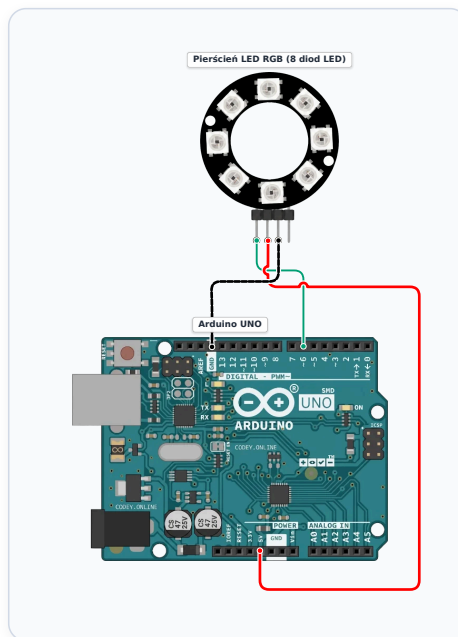
Co to jest?

Pierścień z 8 sprytnymi lampkami. Każda lampka może świecić w dowolnym kolorze. I każdą sterujesz osobno. Tylko jednym przewodem danych! Super do efektów świetlnych.

Tak to podłączysz

- **DI** (data in) do pinu, na przykład D6
- **5V** do 5V
- **GND** do GND
- **DO** zostaw niepodłączone (tam możesz podłączyć kolejny pierścień)

Schemat podłączenia



- Arduino UNO · 5V → Pierścień LED RGB (8 diod LED) · 5V
- Arduino UNO · GND → Pierścień LED RGB (8 diod LED) · GND
- Arduino UNO · D6 → Pierścień LED RGB (8 diod LED) · DI

Co możesz z tym zrobić?

- Lampka tęczęwa
- Pasek świetlny, który pokazuje, jak blisko jest coś
- Kostka do gry ze światła

Wskazówka

Nie ustawiaj w kodzie zbyt dużej jasności. Tak jest lepiej dla twoich oczu i dla Arduino.

Projekty z tą częścią

★ Tęczowy pierścień

★★ Lampka nocna

★★ Elektroniczna kostka

★★★★ Miernik odległości z paskiem świetlnym

Część 14 z 15



Przełącznik 5V

Przełącznik, który może przełączać twoje Arduino

W twoim zestawie: 1x

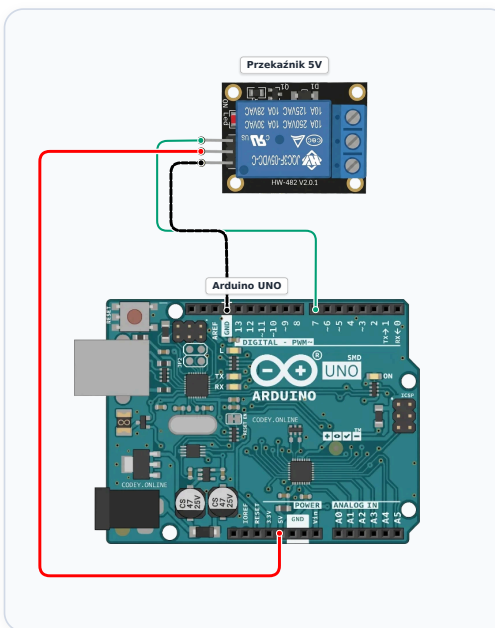
Co to jest?

Przełącznik to przełącznik, który Arduino może włączać i wyłączać. Usłyszysz jego „klik”. Dzięki przełącznikowi Arduino może włączać i wyłączać inne urządzenie, na przykład mały wentylator albo lampkę na baterię.

Tak to podłączysz

- S do pinu, na przykład D7
- + do 5V
- – do GND
- Urządzenie, które chcesz przełączać, podłącz do **COM** i **NO**. NO znaczy: normalnie wyłączone, włączone, gdy przełącznik kliknie.

Schemat podłączenia



● Arduino UNO · GND → Przełącznik 5V · -

● Arduino UNO · 5V → Przełącznik 5V · +

● Arduino UNO · D7 → Przełącznik 5V · S

Co możesz z tym zrobić?

- Wentylator, który włącza się, gdy jest ciepło
- Lampka z timerem
- Mała pompka do roślin

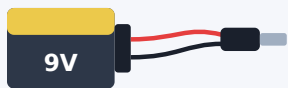
Wskazówka

Bezpieczeństwo przede wszystkim: używaj przełącznika tylko z bateriami albo niskim napięciem (do 12V). Nigdy z gniazdkiem (230V)!

Projekty z tą częścią

★★★★ Sprytny wentylator

Część 15 z 15



Klips do baterii 9V

Niech twój projekt działa bez komputera

W twoim zestawie: 1x

Co to jest?

Tym klipsem podłączysz baterię 9V do Arduino. Wtedy twój projekt działa też bez komputera. To wygodne, gdy chcesz go pokazać albo gdzieś postawić.

Tak to podłączysz

Wepnij baterię w klips i włóż okrągłą wtyczkę do czarnego gniazda zasilania w Arduino. Twój kod nadal jest w Arduino, więc program od razu rusza.

Co możesz z tym zrobić?

- Powiesić czujnik parkowania w garażu
- Postawić stację pogodową na parapecie
- Pokazać swój projekt znajomym

Wskazówka

Odłącz baterię, gdy skończysz. Wtedy wystarczy na dużo dłużej.

Co to jest vibe coding?

Kiedyś najpierw trzeba było uczyć się programowania przez wiele miesięcy. Teraz po prostu mówisz własnymi słowami, co chcesz zrobić. AI pisze kod za ciebie.

To nazywa się **vibe coding**: ty masz pomysł, a AI robi pisanie. Mówisz na przykład: *"Zrób lampkę nocną, która włącza się, gdy robi się ciemno."* Po kilku sekundach kod jest gotowy, razem ze schematem podłączenia.

Dzisiejsza AI bardzo dobrze radzi sobie z programowaniem. Często nawet lepiej i szybciej niż człowiek. Zna wszystkie części, robi mniej literówek i spokojnie wszystko wyjaśnia.

Budowanie projektu: kiedyś i teraz

Programowanie samemu

Sprawdzanie, pisanie, szukanie błędów, próbowanie od nowa...

± 10 godzin

Z Codey

Mówisz, co chcesz, wgrywasz i gotowe.

± 5 minut ↗

Tak to działa



Mówisz, co chcesz zrobić.



Codey pisze kod i rysuje, jak wszystko podłączyć.



Budujesz to samo i wgrywasz kod.



Chcesz coś innego? Powiedz to, a Codey to poprawi.

I jeszcze się przy tym uczysz! Codey wyjaśnia, co robi kod. Dzięki temu coraz lepiej rozumiesz, jak to działa.

Utwórz darmowe konto

Aby pracować z Codey, potrzebujesz konta. To nic nie kosztuje i trwa tylko minutkę. Potem ze swoim zestawem startowym możesz używać Codey bez ograniczeń.

1

Zarejestruj się

Kliknij „Utwórz konto” i wpisz swój adres e-mail oraz hasło. Masz konto Google? Kliknij wtedy „Kontynuuj z Google” i gotowe.

2

Potwierdź swój adres e-mail

Dostaniesz e-mail od Codey Online. Kliknij link w tej wiadomości. Nie widzisz e-maila? Zajrzyj też do spamu.

3

Zaloguj się

Zaloguj się swoim adresem e-mail i hasłem. Potem automatycznie wrócisz do tego samouczka, do następnego kroku.

Mam już konto: zaloguj się →

👉 Wejdź na codey.online/uno, aby utworzyć darmowe konto.

👤 Jesteś jeszcze młody? Utwórz konto razem z tatą, mamą albo inną dorosłą osobą.

Tak działa Codey

Codey to twój pomocnik w programowaniu. Wszystko jest na jednym ekranie. To są najważniejsze przyciski.

The screenshot shows the Codey Online interface for the Arduino Uno R3 Starterkit. The interface includes a chat area on the left, a code editor in the center, and a right panel with instructions and a schematic. Numbered callouts 1 through 6 highlight key features: 1. Chat, 2. Code editor, 3. Compile and upload button, 4. Schematic diagram, 5. Monitor port button, and 6. My kit button.

1 Czat

Tutaj rozmawiasz z Codey. Napisz zwykłymi słowami, co chcesz zrobić albo zmienić.

2 Kod

Tutaj jest kod, który Codey pisze dla ciebie. Ty też możesz tu pisać.

3 Kompiluj i wgraj na płytkę

Jeden przycisk do wszystkiego: Codey sprawdza twój kod, przygotowuje go (kompiluje) i wysyła przez kabel USB do twojego Arduino (wgrywa). Jest błąd? Codey często sam go naprawia.

4 Schemat połączeń

Pokazuje, który przewód gdzie ma być podłączony. Widzisz go na czacie; kliknij go, aby zobaczyć większy obraz.

5 Monitor portu szeregowego

Tutaj widzisz wiadomości, które wysyła twoje Arduino, na przykład temperaturę albo odległość.

6 Mój zestaw

Wszystkie części z twojego zestawu startowego. Codey używa tylko rzeczy, które masz.

Dobrze wiedzieć: ze swoim zestawem startowym możesz używać Codey bez limitu w czatach swojego zestawu. Możesz zapytać Codey o wszystko, także "co robi ta linijka?" albo "dlaczego to nie działa?".

Podłączanie i wgrywanie

Twój kod jest gotowy? Teraz trafi do Arduino. Zrobisz to w trzech krokach.

Ważne: użyj **Google Chrome** lub **Microsoft Edge** na komputerze. Tylko te przeglądarki mogą rozmawiać z Arduino przez USB. Safari, Firefox i większość telefonów (jeszcze) tego nie potrafią.

1 Podłącz Arduino

Włóż kabel USB do Arduino i do komputera. Zapali się zielona lampka (ON).

2 Kliknij Kompiluj i wgraj na płytkę

Codey sprawdza twój kod i przygotowuje go (kompiluje). Za pierwszym razem może to potrwać minutkę.

3 Wybierz swoje Arduino w oknie portów

Pojawi się okno z portami. Wybierz swoje Arduino i kliknij Połącz. Lampki RX i TX chwilę zamigają: twój kod trafia do środka (wgrywanie). Gotowe!

Jak nazywa się moje Arduino w oknie portów?

To zależy od komputera. Widzisz kilka opcji? Wyjmij na chwilę kabel i zobacz, która zniknie. To jest twoje Arduino.

Windows

Port **COM** z numerem, na przykład `Arduino Uno (COM4)` albo `USB-SERIAL CH340 (COM5)`. COM1 prawie nigdy nie jest twoim Arduino.

Mac

Coś z `usbmodem` albo `usbserial`, na przykład `/dev/cu.usbmodem14101`, `cu.usbserial-1420` albo `cu.wchusbserial1420`. Zawsze wybieraj wersję `cu`.

Linux

`/dev/ttyACM0` albo `/dev/ttyUSB0`. Nie masz dostępu? Dodaj siebie do grupy `dialout` i zaloguj się ponownie.

Chromebook

Chrome pokaże nazwę urządzenia, na przykład `Arduino Uno`, `USB2.0-Serial` albo `USB Serial`.

Monitor szeregowy: niektóre projekty wysyłają wiadomości do komputera, na przykład temperaturę. Zobaczysz je, gdy klikniesz w Codey Monitor portu szeregowego. Tak widzisz, co twoje Arduino "myśli".

Nie działa? Bez paniki!

▼ Nie widzę mojego Arduino w oknie portów.

Spróbuj innego kabla USB. Niektóre kable tylko ładują i nie wysyłają danych. Spróbuj też innego portu USB w komputerze.

▼ Widzę CH340 albo USB2.0-Serial, ale wgrywanie nie działa (Windows/Mac).

Twoje Arduino używa wtedy układu CH340. Zainstaluj raz darmowy sterownik CH340 (wyszukaj "CH340 driver") i uruchom komputer ponownie.

▼ Wgrywanie przerywa się w połowie.

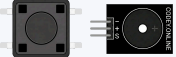
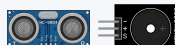
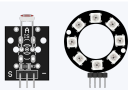
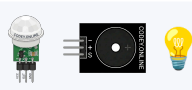
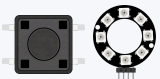
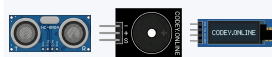
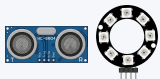
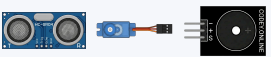
Zamknij inne programy, które używają Arduino, na przykład Arduino IDE albo drugą kartę Codey. Wyjmij na chwilę kabel, włóż go z powrotem i spróbuj jeszcze raz.

▼ Działa, ale mój układ nic nie robi.

Sprawdź przewody według schematu podłączenia. Czy 5V i GND są dobrze podłączone? Potem zapytaj Codey: "Moja LED się nie świeci, co to może być?"

Do dzieła!

Wybierz projekt i kliknij ZRÓB TO. Codey przygotuje dla Ciebie kod i schemat połączeń. Zaczynaj od projektu z jedną gwiazdką ★, jeśli jeszcze nigdy nie robiłeś nic z Arduino.

 ★ ⌚ 10 minut Twój pierwszy układ na breadboardzie Poćwicz z breadboardem: zapal diodę LED, a potem niech miga.	 ★ ⌚ 10 minut Migacz Twój pierwszy projekt: spraw, żeby LED migała.	 ★ ⌚ 15 minut Sygnalizacja świetlna Trzy diody LED działają jak prawdziwe światła.	 ★ ⌚ 15 minut Przycisk światła Naciśnij przycisk, a lampka się włączy. Jeszcze raz: wyłącz się.
 ★ ⌚ 10 minut Dzwonek do drzwi Naciśnij przycisk i usłysz wesołe ding-dong.	 ★ ⌚ 10 minut Tęczowy pierścień Pierścień LED kręci się we wszystkich kolorach tęczy.	 ★★ ⌚ 20 minut Mini-stacja pogody Temperatura i wilgotność powietrza na twoim własnym ekraniku.	 ★★ ⌚ 20 minut Czujnik parkowania Im bliżej, tym szybciej piszczy. Tak jak w aucie.
 ★★ ⌚ 15 minut Lampka nocna Robi się ciemno? Wtedy lampka sama się włącza.	 ★★ ⌚ 20 minut Alarm ruchu Ktoś przechodzi? Wtedy włącza się alarm!	 ★★ ⌚ 20 minut Szlaban Naciśnij przycisk, a szlaban się otworzy.	 ★★ ⌚ 25 minut Gra na refleks Jak szybki jesteś? Zmierz swój czas reakcji.
 ★★ ⌚ 15 minut Elektroniczna kostka Naciśnij przycisk i rzuć kostką ze światła.	 ★★★ ⌚ 30 minut Czujnik parkowania deluxe Pika i pokazuje odległość w centymetrach na małym ekranie.	 ★★★ ⌚ 25 minut Miernik odległości z paskiem świetlnym Im bliżej, tym więcej lampek: zielone, pomarańczowe, czerwone.	 ★★★ ⌚ 30 minut Sprytny wentylator Za ciepło? Wtedy przekaźnik włącza wentylator.
 ★★★ ⌚ 30 minut Automatyczny szlaban Jedzie auto? Wtedy szlaban sam się otwiera.			

Masz własny pomysł? Jeszcze lepiej! Otwórz Codey i powiedz, co chcesz zrobić. Na przykład: "Zrób lampkę, która miga szybciej, gdy robi się ciepło."

★ Początkujący ⌚ 10 minut

Twój pierwszy układ na breadboardzie

To ćwiczenie pokaże ci, jak działa breadboard. Zbudujesz układ z diodą LED i rezystorem na breadboardzie. Najpierw zapalisz ją paskiem 5V, bez kodu. Potem sprawisz, że będzie migać z pin 13.

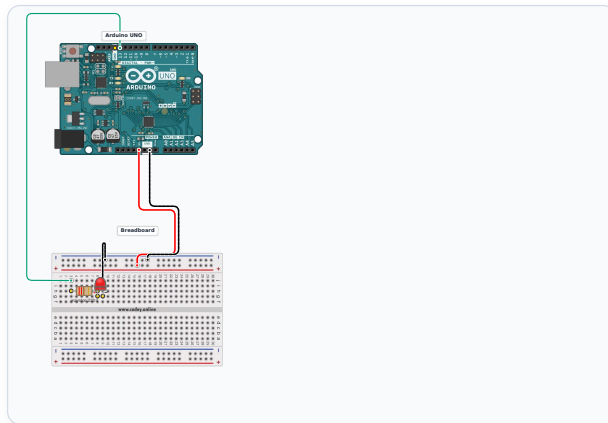
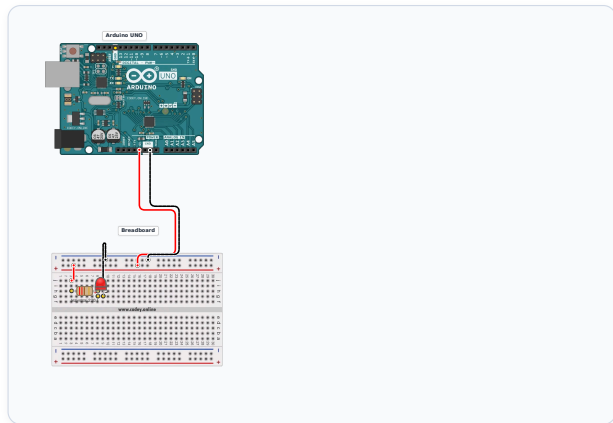
Schemat podłączenia

Krok A: zapal diodę LED (bez kodu)

Czerwony przewód idzie od paska + do rzędu 3. Prąd płynie tak: 5V → pasek + → rezystor → LED → pasek - → GND. Dioda LED świeci? Super! Dobrze używasz breadboarda!

Krok B: spraw, aby dioda LED migiała z pin 13

Wyjmij czerwony przewód między paskiem + a rzędem 3. Zamiast niego włóż przewód od pin 13 do rzędu 3. Teraz Arduino decyduje, kiedy dioda LED dostaje prąd.



Czego potrzebujesz?

 1× Arduino UNO	 1× Breadboard	 1× LED (lampka)
 Czerwony, czerwony, brązowy = 220 Ω 1× Rezystor 220 Ω	 Kabel USB + Przewody połączeniowe	

Tak to zbudujesz

- Połącz 5V z Arduino z paskiem + u góry breadboarda (czerwony przewód).
- Połącz GND z Arduino z paskiem - obok (czarny przewód). Teraz masz dwa paski zasilania.
- Włóż rezystor 220 Ω od otworu H3 do otworu H8. Leży wtedy w dwóch różnych rzędach.
- Włóż dłuższą nóżkę diody LED do G8 (ten sam rząd co rezystor), a krótszą nóżkę do G9.
- Połącz otwór J9 z paskiem - (czarny przewód). Tak prąd może wrócić do GND.
- Krok A: połącz pasek + z otworem J3 (czerwony przewód). Dioda LED się zapali!
- Krok B: wyjmij ten czerwony przewód i połącz pin 13 z otworem J3 (zielony przewód).
- Niech Codey zrobi kod: kliknij ZRÓB TO albo wpisz samodzielnie na czacie: "Niech dioda LED na pin 13 miga co sekundę." Potem kliknij Kompiluj i wgraj na płytke.

Kod

```
// Twój pierwszy układ na breadboardzie
// Dioda LED na breadboardzie włącza się i wyłącza. Dostaje prąd z pin 13.

const int LED_PIN = 13; // Zielony przewód idzie od pin 13 do breadboarda

void setup() {
  pinMode(LED_PIN, OUTPUT); // Pin 13 jest wyjściem: daje prąd
}

void loop() {
  digitalWrite(LED_PIN, HIGH); // LED włączona
  delay(1000);                // Czekaj 1 sekundę
  digitalWrite(LED_PIN, LOW);  // LED wyłączona
  delay(1000);                // Czekaj 1 sekundę
}
```

Co robi kod?

- Pin 13 jest wyjściem. Przy **HIGH** daje 5V, a przy **LOW** nie daje nic.
- W **loop()** włączamy diodę LED, czekamy 1 sekundę, wyłączamy ją i znowu czekamy.
- Prąd płynie przez breadboard: pin 13 → rząd 3 → rezystor → rząd 8 → LED → rząd 9 → pasek - → GND.

Spróbuj...

- Spraw, aby dioda LED migiała dwa razy szybciej
- Dodaj drugą diodę LED na breadboardzie, która miga na zmianę
- Spraw, aby dioda LED zamigiała trzy razy krótko, a potem chwilę była wyłączona

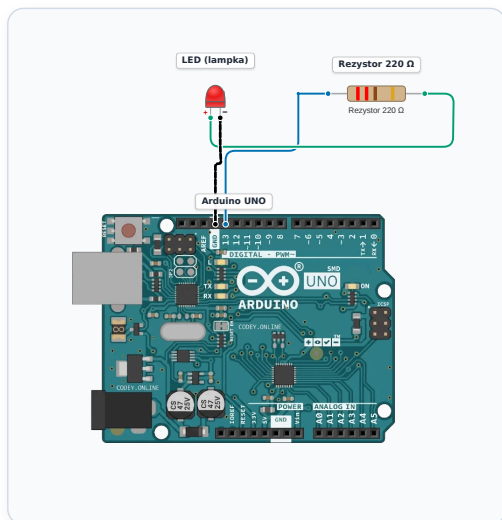
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/breadboard-led i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★ Początkujący ⌚ 10 minut

Migacz

Twój pierwszy projekt! LED wciąż się włącza i wyłącza, jak migacz. Tak uczysz się, jak Arduino włącza i wyłącza wyjście.

Schemat podłączenia



Czego potrzebujesz?



Tak to zbudujesz

1. Włóż LED do breadboardu, nóżkami w dwa różne rzędy.
2. Włóż rezystor 220 Ω od rzędu z długą nóżką do innego rzędu.
3. Połącz ten inny rząd przewodem z pin 13 Arduino.
4. Połącz krótką nóżkę przewodem z GND.

Kod

```
// Migacz
// LED wciąż się włącza i wyłącza, jak migacz.

const int LED_PIN = 13; // LED jest na pinie 13 (przez rezystor 220 ohm)
const int CZAS_CZEKANIA = 500; // Jak długo LED świeci lub nie świeci (w milisekundach)

void setup() {
  // setup() działa raz, gdy Arduino startuje
  pinMode(LED_PIN, OUTPUT); // Pin 13 to wyjście: daje prąd
}

void loop() {
  // loop() działa wciąż od nowa, na zawsze
  digitalWrite(LED_PIN, HIGH); // LED włączona
  delay(CZAS_CZEKANIA); // Chwilę czekamy
  digitalWrite(LED_PIN, LOW); // LED wyłączona
  delay(CZAS_CZEKANIA); // Chwilę czekamy
}
```

Co robi kod?

- W `setup()` mówimy, że pin 13 jest wyjściem.
- W `loop()` włączamy LED, chwilę czekamy, wyłączamy ją i znowu czekamy.
- Potem `loop()` po prostu zaczyna od nowa. Tak LED cały czas miga.

Spróbuj...

- Spraw, żeby LED migała dwa razy szybciej
- Zrób z tego sygnał SOS: krótko-krótko-krótko, długo-długo-długo, krótko-krótko-krótko
- Spraw, żeby LED powoli świeciła mocniej i słabiej, jakby oddychała

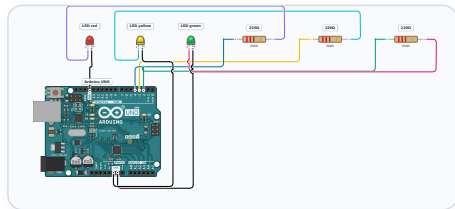
🚀 Zbuduj ten projekt z Codey: [otwórz codey.online/pl/starter-kits/uno/projects/blink](https://codey.online/pl/starter-kits/uno/projects/blink) i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★ Początkujący ⌚ 15 minut

Sygnalizacja świetlna

Zbudujesz prawdziwą sygnalizację świetlną z czerwoną, żółtą i zieloną diodą LED. Zapalają się po kolei, tak jak na ulicy.

Schemat podłączenia



Czego potrzebujesz?



1x Arduino UNO



3x LED (lampka)

Czerwony, czerwony, brązowy = 220 Ω
3x Rezystor 220 ΩKabel USB +
Przewody
połączeniowe

Tak to zbudujesz

1. Włóż trzy diody LED obok siebie w breadboard.
2. Daj każdej diodzie LED własny rezystor 220 Ω do jej pinu: czerwona do 4, żółta do 3, zielona do 2.
3. Połącz wszystkie krótkie nóżki z paskiem minusa (–) na breadboard.
4. Połącz pasek minusa z GND w Arduino.

Kod

```
// Sygnalizacja świetlna
// Trzy diody LED robią dokładnie to, co prawdziwe światła: zielone, pomarańczowe, czerwone.
```

```
const int CZERWONY = 4;    // Czerwona LED na pin 4
const int POMARANCZOWY = 3; // Żółta/pomarańczowa LED na pin 3
const int ZIELONY = 2;     // Zielona LED na pin 2
```

```
void setup() {
  pinMode(CZERWONY, OUTPUT);
  pinMode(POMARANCZOWY, OUTPUT);
  pinMode(ZIELONY, OUTPUT);
}
```

```
// Włącz jedną lampkę, a dwie pozostałe wyłącz
void ustawSwiatlo(int włączona) {
  digitalWrite(CZERWONY, włączona == CZERWONY ? HIGH : LOW);
  digitalWrite(POMARANCZOWY, włączona == POMARANCZOWY ? HIGH : LOW);
  digitalWrite(ZIELONY, włączona == ZIELONY ? HIGH : LOW);
}
```

```
void loop() {
  ustawSwiatlo(ZIELONY);    // Zielone: możesz jechać
  delay(5000);
  ustawSwiatlo(POMARANCZOWY); // Pomarańczowe: uwaga, zaraz czerwone
  delay(2000);
  ustawSwiatlo(CZERWONY);   // Czerwone: stop!
  delay(5000);
}
```

Co robi kod?

- Każda dioda LED ma własny pin: czerwona na 4, żółta na 3 i zielona na 2.
- Funkcja `ustawSwiatlo()` włącza jedną lampkę, a dwie pozostałe wyłącza.
- W `loop()` przechodzimy od zielonego do pomarańczowego, a potem do czerwonego, z krótką przerwą pomiędzy.

Spróbuj...

- Spraw, aby pomarańczowa lampka mignęła trzy razy, zanim zapali się czerwona
- Dodaj przycisk: gdy go naciśniesz, światło szybciej przeskoczy na czerwone (przycisk dla pieszych)
- Zrób tryb nocny, w którym miga tylko pomarańczowa lampka

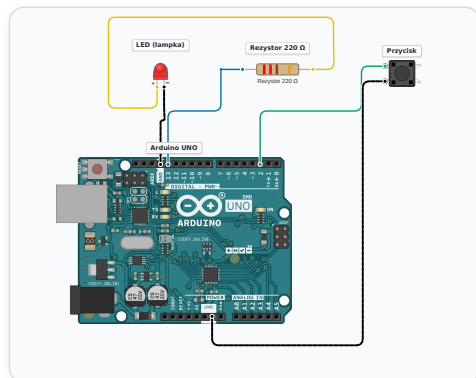
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/traffic-light i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★ Początkujący ⌚ 15 minut

Przycisk światła

Zrobisz przycisk światła. Naciśnij raz: LED się włącza. Naciśnij jeszcze raz: LED się wyłącza. Tak nauczysz się, jak Arduino „czuje” przycisk.

Schemat podłączenia



Czego potrzebujesz?



Tak to zbudujesz

1. Włóż przycisk przez szczelinę w breadboard.
2. Połącz jedną nóżkę z pin 2, a nóżkę po drugiej stronie z GND.
3. Podłącz LED z rezystorem 220 Ω do pin 13. Krótka nóżka idzie do GND.

Kod

```
// Przycisk światła
// Naciśnij przycisk raz: LED się włącza. Jeszcze raz: LED się wyłącza.

const int PRZYCISK_PIN = 2; // Przycisk jest na pinie 2 (druga strona do GND)
const int LED_PIN = 13; // LED jest na pinie 13 (przez rezystor)

bool ledWlaczona = false; // Pamięta, czy LED jest teraz włączona czy wyłączona
bool poprzedniPrzycisk = HIGH; // Jaki był przycisk poprzednim razem

void setup() {
  pinMode(PRZYCISK_PIN, INPUT_PULLUP); // Wejście z wbudowanym rezystorem: puszczone = HIGH, wciśnięty = LOW
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  bool przycisk = digitalRead(PRZYCISK_PIN);

  // Reaguj tylko w chwili, gdy naciskasz przycisk
  if (poprzedniPrzycisk == HIGH && przycisk == LOW) {
    ledWlaczona = !ledWlaczona; // Odwróć: włączona staje się wyłączona, wyłączona staje się włączona
    digitalWrite(LED_PIN, ledWlaczona ? HIGH : LOW);
    delay(50); // Krótko poczekaj na „drganie” przycisku
  }

  poprzedniPrzycisk = przycisk;
}
```

Co robi kod?

- Przycisk jest na pinie 2 z `INPUT_PULLUP`. Gdy go nie naciskasz, pin ma HIGH. Gdy naciskasz, ma LOW.
- Pamiętamy, czy LED jest włączona czy wyłączona w `ledWlaczona`.
- Tylko w chwili, gdy naciskasz przycisk, odwracamy `ledWlaczona`.

Spróbuj...

- Niech LED miga zamiast świecić cały czas
- Niech LED sama wyłączy się po 10 sekundach
- Policz, ile razy naciśnięto przycisk, i pokaż to w monitorze szeregowym

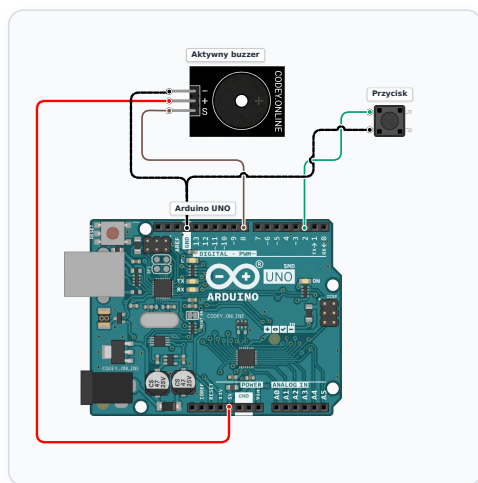
 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/light-switch i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★ Początkujący ⌚ 10 minut

Dzwonek do drzwi

Zrobisz własny dzwonek do drzwi. Naciśnij przycisk, a brzęczyk powie "ding-dong"!

Schemat podłączenia



Czego potrzebujesz?



1× Arduino UNO



1× Przycisk



1× Aktywny buzzer

Kabel USB +
Przewody
połączeniowe

Tak to zbudujesz

1. Włóż przycisk nad rowkiem breadboard: jedna nóżka do pin 2, nóżka po drugiej stronie do GND.
2. Podłącz brzęczyk: S do pin 8, + do 5V i – do GND.

Kod

```
// Dzwonek do drzwi
// Naciśnij przycisk, a brzęczyk zagra wesołe "ding-dong".

const int PRZYCISK_PIN = 2; // Przycisk na pin 2 (druga strona do GND)
const int BRZECZYK_PIN = 8; // Aktywny brzęczyk: S na pin 8

void setup() {
  pinMode(PRZYCISK_PIN, INPUT_PULLUP); // Puszczony = HIGH, wciśnięty = LOW
  pinMode(BRZECZYK_PIN, OUTPUT);
}

// Brzęczyk pika przez chwilę
void pik(int jakDługo) {
  digitalWrite(BRZECZYK_PIN, HIGH);
  delay(jakDługo);
  digitalWrite(BRZECZYK_PIN, LOW);
}

void loop() {
  if (digitalRead(PRZYCISK_PIN) == LOW) { // Ktoś naciska dzwonek!
    pik(300); // "Ding"
    delay(150);
    pik(600); // "Dong"
    delay(1000); // Chwilę czekamy, żeby dzwonek nie dzwonił bez końca
  }
}
```

Co robi kod?

- Przycisk jest na pin 2, a brzęczyk na pin 8.
- Funkcja `pik()` włącza brzęczyk na chwilę, a potem znów go wyłącza.
- Gdy przycisk jest wciśnięty, usłyszymy krótki i długi pik.

Spróbuj...

- Zagraj wesołą melodyjkę z pięciu piknięć
- Niech LED też miga, gdy ktoś dzwoni
- Policz, ile razy ktoś zadzwonił, i pokaż to w monitorze szeregowym

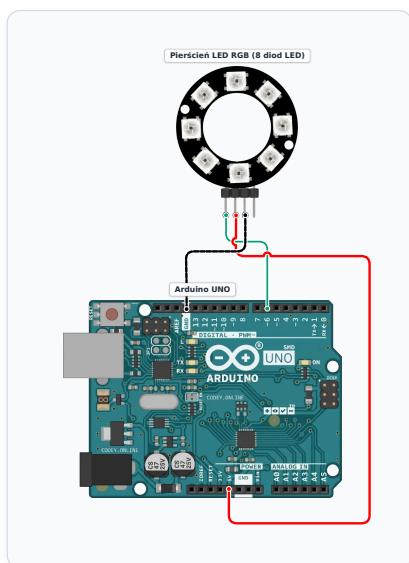
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/doorbell i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★ Początkujący ⌚ 10 minut

Tęczowy pierścień

Z pierścieniem LED zrobisz kręcącą się tęczę. Każda lampka dostaje inny kolor, a wszystkie kolory przesuwają się w kółko.

Schemat podłączenia



Czego potrzebujesz?



1× Arduino UNO



1× Pierścień LED RGB (8 diod LED)



Kabel USB + Przewody połączeniowe

Tak to zbudujesz

1. Podłącz pierścień LED: DI do pin 6, 5V do 5V i GND do GND.
2. To wszystko! DO zostaw niepodłączone.

Kod

```
// Tęczowy pierścień
// Pierścień LED kręci kolorami tęczy w kółko.

#include <Adafruit_NeoPixel.h>

const int PIN_PIERSCIENIA = 6; // DI pierścienia jest na pinie 6
const int LICZBA_LED = 8; // Pierścień ma 8 lampek
const int PREDKOSC = 20; // Im mniej, tym szybciej się kręci

Adafruit_NeoPixel pierscien(LICZBA_LED, PIN_PIERSCIENIA, NEO_GRB + NEO_KHZ800);

void setup() {
  pierscien.begin();
  pierscien.setBrightness(40); // Nie za jasno: 0 = wył., 255 = superjasno
  pierscien.show();
}

void loop() {
  // Przesuwaj tęczę ciągle o kawałek
  for (long kolor = 0; kolor < 65536; kolor += 512) {
    for (int i = 0; i < LICZBA_LED; i++) {
      // Każda lampka dostaje trochę inny kolor
      long kolorLampki = kolor + (i * 65536L / LICZBA_LED);
      pierscien.setPixelColor(i, pierscien.gamma32(pierscien.ColorHSV(kolorLampki)));
    }
    pierscien.show();
    delay(PREDKOSC);
  }
}
```

Co robi kod?

- Używamy biblioteki `Adafruit_NeoPixel`, aby sterować pierścieniem.
- `ColorHSV()` tworzy kolor z tęczy. Każda lampka dostaje kolor kawałek dalej w tęczy.
- Gdy cały czas przesuwamy kolor o mały kawałek, wygląda to tak, jakby tęczę kręciła się w kółko.

Spróbuj...

- Niech tęczę kręci się dwa razy szybciej
- Zrób z tego światło policyjne, które miga na zmianę na czerwono i niebiesko
- Niech jedna lampka kręci się w kółko, jak światełko, które biega dookoła

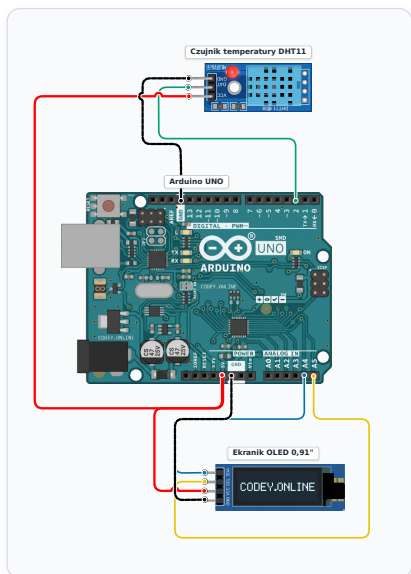
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/rainbow-ring i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★★ Początkujący plus ⌚ 20 minut

Mini-stacja pogody

Zrobisz mini-stację pogody. DHT11 mierzy temperaturę i wilgotność powietrza, a ekranik OLED je pokazuje.

Schemat podłączenia



Czego potrzebujesz?



1x Arduino UNO



1x Czujnik temperatury DHT11



1x Ekranik OLED 0,91"



Kabel USB + Przewody połączeniowe

Tak to zbudujesz

1. Podłącz DHT11: DAT do pin 2, VCC do 5V i GND do GND.
2. Podłącz ekran OLED: SDA do A4, SCL do A5, VCC do 5V i GND do GND.
3. Wskazówka: dwie części potrzebują 5V i GND. Użyj pasków + i - na swoim breadboardzie.

Kod

```
// Mini-stacja pogody
// DHT11 mierzy temperaturę i wilgotność powietrza. Ekranik OLED to pokazuje.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2;          // DAT z DHT11 jest na pin 2
DHT dht(DHT_PIN, DHT11);

Adafruit_SSD1306 ekran(128, 32, &Wire, -1); // OLED 128 x 32 piksele (SDA = A4, SCL = A5)

void setup() {
  Serial.begin(9600);
  dht.begin();
  if (!ekran.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // 0x3C to adres ekranu
    Serial.println(F("Nie ma ekranu. Sprawdź SDA i SCL."));
    while (true); // Stop: bez ekranu to nie ma sensu
  }
  ekran.setTextColor(SSD1306_WHITE);
}

void loop() {
  float temperatura = dht.readTemperature(); // W stopniach Celsius

  float wilgotnosc = dht.readHumidity();      // W procentach

  ekran.clearDisplay();
  ekran.setTextSize(2);
  ekran.setCursor(0, 0);

  if (isnan(temperatura) || isnan(wilgotnosc)) { // Pomiar się nie udał?
    ekran.setTextSize(1);
    ekran.print(F("Brak DHT11"));
  } else {
    ekran.print(temperatura, 1);
    ekran.print((char)247); // Znak stopnia
    ekran.println(F("C"));
    ekran.print(wilgotnosc, 0);
    ekran.print(F("% wilg."));

    Serial.print(F("Temperatura: "));
    Serial.print(temperatura);
    Serial.print(F(" C Wilg.: "));
    Serial.print(wilgotnosc);
    Serial.println(F("%"));
  }

  ekran.display(); // Dopiero teraz wszystko naprawdę trafia na ekran
  delay(2000);    // DHT11 może mierzyć tylko raz na 2 sekundy
}
```

Co robi kod?

- DHT11 jest na pinie 2. Za pomocą `readTemperature()` i `readHumidity()` odczytujemy wartości.
- Ekran OLED rozmawia przez SDA (A4) i SCL (A5). Za pomocą `print()` pokazujemy tekst na ekranie.
- Co 2 sekundy pojawia się nowy pomiar. Nie uda się zmierzyć? Wtedy zobaczysz komunikat.

Spróbuj...

- Pokaż też temperaturę w Fahrenheit
- Włącz LED, gdy jest cieplej niż 25 stopni
- Zapamiętaj najwyższą temperaturę z dzisiaj i też ją pokaż

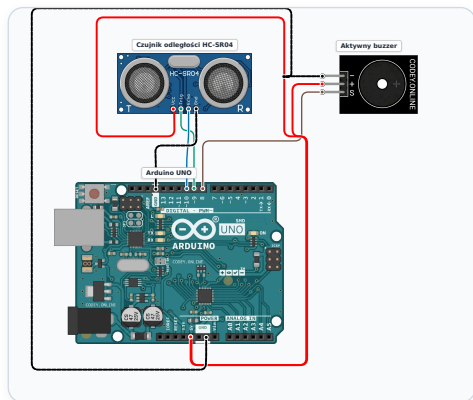
🔧 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/weather-station i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

☆☆ Początkujący plus ⌚ 20 minut

Czujnik parkowania

Zbudujesz czujnik parkowania, tak jak w aucie. Im bliżej jesteś czujnika, tym szybciej piszczy brzęczyk. Bardzo blisko? Wtedy słysząc jeden długi dźwięk.

Schemat podłączenia



Czego potrzebujesz?



Tak to zbudujesz

1. Podłącz HC-SR04: Trig do pin 9, Echo do pin 10, Vcc do 5V i Gnd do GND.
2. Podłącz brzęczyk: S do pin 8, + do 5V i – do GND.
3. Użyj pasków + i – na swoim breadboard do 5V i GND.

Kod

```
// Czujnik parkowania
// Im bliżej jesteś czujnika, tym szybciej piszczy brzęczyk. Tak jak
// w aucie!

const int TRIG_PIN = 9;    // Trig z HC-SR04 na pin 9
const int ECHO_PIN = 10;   // Echo z HC-SR04 na pin 10
const int BRZECZYK_PIN = 8; // S brzęczyka na pin 8

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BRZECZYK_PIN, OUTPUT);
}

// Mierzy odległość w centymetrach
long zmierzOdleglosc() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);    // Wyślij krótki dźwięk (którego
  // nie słyszysz)
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long czas = pulseIn(ECHO_PIN, HIGH, 30000); // Ile trwa powrót
  // echa?
  if (czas == 0) return 999;              // Brak echa: nic nie
  // ma w pobliżu
  return czas / 58;                       // Przelicz czas na
  // centymetry
}

void loop() {
  long odleglosc = zmierzOdleglosc();
  Serial.print(F("Odległość: "));
  Serial.print(odleglosc);
  Serial.println(F(" cm"));

  if (odleglosc < 5) {
    digitalWrite(BRZECZYK_PIN, HIGH); // Bardzo blisko: jeden
    // długi ton
    delay(100);
  } else if (odleglosc < 50) {
    digitalWrite(BRZECZYK_PIN, HIGH); // Krótkie piknięcie
    delay(50);
    digitalWrite(BRZECZYK_PIN, LOW);
    delay(odleglosc * 10);             // Bliżej = krótsza przerwa
    // = szybsze pieszczanie
  } else {
    digitalWrite(BRZECZYK_PIN, LOW); // Daleko: cisza
    delay(100);
  }
}
```

Co robi kod?

- `zmierzOdleglosc()` wysyła dźwięk i mierzy, jak długo trwa powrót echa. Potem przeliczamy to na centymetry.
- Dalej niż 50 cm: brzęczyk jest cicho.
- Bliżej: przerwa między piknięciami robi się krótsza. Poniżej 5 cm piszczy bez przerwy.

Spróbuj...

- Niech czujnik parkowania zacznie piszczeć już od 1 meter
- Dodaj czerwoną LED, która zaświeci, gdy jest za blisko
- Pokaż odległość w monitorze szeregowym, z paseczkiem z gwiazdek

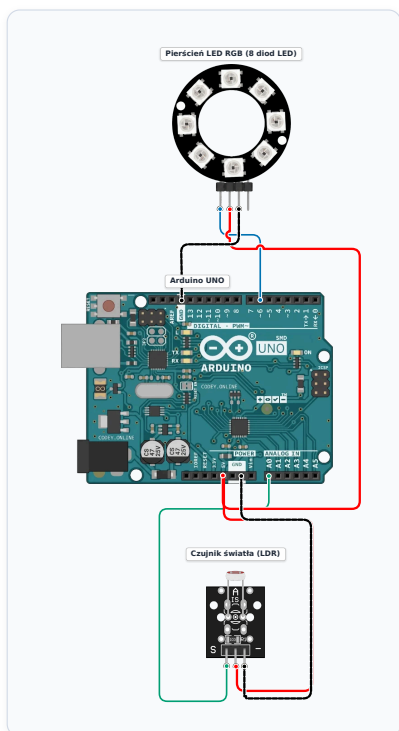
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/parking-sensor i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★★ Początkujący plus ⌚ 15 minut

Lampka nocna

Zrobisz sprytną lampkę nocną. Gdy robi się ciemno, pierścień LED świeci delikatnie. Gdy robi się jasno, znów gaśnie.

Schemat podłączenia



Czego potrzebujesz?



1x Arduino UNO



1x Czujnik światła (LDR)



1x Pierścień LED RGB (8 diod LED)



Kabel USB + Przewody połączeniowe

Tak to zbudujesz

1. Podłącz czujnik światła: S do A0, + do 5V i – do GND.
2. Podłącz pierścień LED: DI do pin 6, 5V do 5V i GND do GND.
3. Otwórz monitor portu szeregowego i zobacz, jaką liczbę widzisz przy świetle i w ciemności. Dopasuj do tego `GRANICA_CIEMNOSCI`.

Kod

```
// Lampka nocna
// Robi się ciemno? Wtedy pierścień LED świeci delikatnie. Robi się jasno? Wtedy
gaśnie.

#include <Adafruit_NeoPixel.h>

const int PIN_SWIATLA = A0; // S czujnika światła do A0
const int PIN_RINGU = 6; // DI pierścienia LED do pin 6
const int LICZBA_LED = 8;

// Przy tej wartości uznajemy, że jest „ciemno”. Zobaczyć w monitorze portu
szeregowego, jaką liczbę widzisz
// przy świetle i w ciemności, i wpisz tutaj liczbę pomiędzy nimi.
const int GRANICA_CIEMNOSCI = 500;

Adafruit_NeoPixel pierscien(LICZBA_LED, PIN_RINGU, NEO_GRB + NEO_KHZ800);

void setup() {
  Serial.begin(9600);
  pierscien.begin();
  pierscien.setBrightness(60);
  pierscien.show();
}

void loop() {
  int swiatlo = analogRead(PIN_SWIATLA); // Liczba od 0 do 1023
  Serial.print(F("Czujnik światła: "));
  Serial.println(swiatlo);

  // Przy tym czujniku liczba rośnie, gdy robi się ciemniej.
  // U ciebie działa odwrotnie? Zmień wtedy > na <.
  bool ciemno = swiatlo > GRANICA_CIEMNOSCI;

  for (int i = 0; i < LICZBA_LED; i++) {
    if (ciemno) {
      pierscien.setPixelColor(i, pierscien.Color(255, 140, 40)); // Ciepłe żółte
    } else {
      pierscien.setPixelColor(i, 0); // Wyłączone
    }
  }
  pierscien.show();
  delay(200);
}
```

Co robi kod?

- Czujnik światła jest podłączony do A0. `analogRead()` daje liczbę od 0 do 1023.
- Czy liczba jest większa niż `GRANICA_CIEMNOSCI`? Wtedy uznajemy, że jest ciemno.
- W ciemności wszystkie lampki świecą ciepłym żółtym kolorem. Inaczej gasną.

Spróbuj...

- Niech lampka świeci coraz mocniej, im ciemniej się robi
- Niech kolor powoli się zmienia, jak spokojna lampka nastrojowa
- Niech lampka po 1 minucie sama zgaśnie

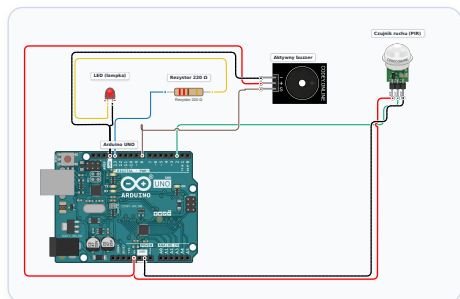
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/night-light i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

☆☆ Początkujący plus ⌚ 20 minut

Alarm ruchu

Zbudujesz alarm antywłamaniowy. Czujnik ruchu widzi kogoś? Wtedy brzęczyk piszczy i miga czerwona LED.

Schemat podłączenia



Czego potrzebujesz?



Tak to zbudujesz

1. Podłącz czujnik PIR: OUT do pin 2, VCC do 5V i GND do GND.
2. Podłącz brzęczyk: S do pin 8, + do 5V i – do GND.
3. Podłącz czerwoną LED z rezystorem 220 Ω do pin 13. Krótsza nóżka idzie do GND.

Kod

```
// Alarm ruchu
// Ktoś przechodzi? Wtedy czerwona LED miga, a brzęczyk piszczy.

const int PIR_PIN = 2;    // OUT czujnika ruchu na pin 2
const int BRZECZYK_PIN = 8; // S brzęczyka na pin 8
const int LED_PIN = 13;   // Czerwona LED na pin 13 (przez rezystor)

void setup() {
  Serial.begin(9600);
  pinMode(PIR_PIN, INPUT);
  pinMode(BRZECZYK_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.println(F("Czujnik startuje... stój spokojnie!"));
  delay(10000); // Czujnik potrzebuje chwili, aby przyzwyczaić się do pokoju
  Serial.println(F("Alarm jest włączony."));
}

void loop() {
  if (digitalRead(PIR_PIN) == HIGH) { // Wykryto ruch!
    Serial.println(F("Ruch! ALARM!"));
    for (int i = 0; i < 5; i++) { // 5 razy piszczy i miga
      digitalWrite(LED_PIN, HIGH);
      digitalWrite(BRZECZYK_PIN, HIGH);
      delay(150);
      digitalWrite(LED_PIN, LOW);
      digitalWrite(BRZECZYK_PIN, LOW);
      delay(150);
    }
  }
  delay(100);
}
```

Co robi kod?

- Czujnik PIR jest podłączony do pin 2. Gdy widzi ruch, pin jest HIGH.
- Przy starcie czekamy 10 sekund. Dzięki temu czujnik może przyzwyczaić się do pokoju.
- Gdy jest ruch, brzęczyk piszczy, a LED miga 5 razy.

Spróbuj...

- Dodaj przycisk, aby włączać i wyłączać alarm
- Policz, ile razy był ruch, i pokaż to w monitorze szeregowym
- Niech alarm za drugim razem brzmi dłużej i głośniej

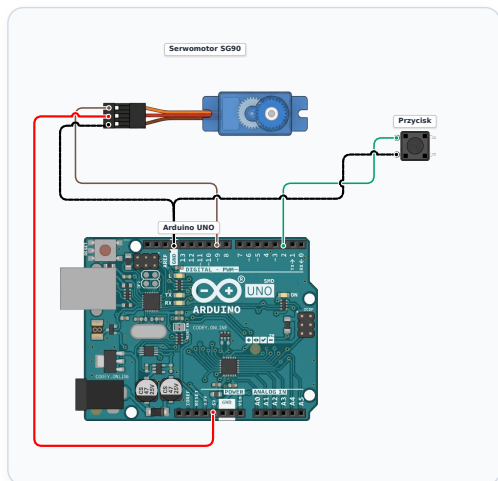
 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/motion-alarm i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

☆☆ Początkujący plus ⌚ 20 minut

Szlaban

Budujesz szlaban, taki jak przy parkingu. Naciśnij przycisk: serwo podnosi szlaban, chwilę czeka i znów go zamyka.

Schemat podłączenia



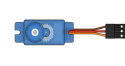
Czego potrzebujesz?



1x Arduino UNO



1x Przycisk



1x Serwomotor SG90

Kabel USB +
Przewody
połączeniowe

Tak to zbudujesz

1. Włóż przycisk nad rowkiem w breadboard: jedna nóżka do pin 2, a nóżka po drugiej stronie do GND.
2. Podłącz serwo: pomarańczowy przewód do pin 9, czerwony do 5V, a brązowy do GND.
3. Przyklej słomkę albo patyczek po lodach do ramienia serwa: to jest twój szlaban!

Kod

```
// Szlaban
// Naciśnij przycisk: szlaban (serwo) podnosi się, chwilę czeka
i znów się zamyka.

#include <Servo.h>

const int PRZYCISK_PIN = 2; // Przycisk na pin 2 (druga strona
do GND)
const int SERVO_PIN = 9;    // Pomarańczowy/żółty przewód serwa
na pin 9

const int ZAMKNIĘTY = 0;    // Kat serwa, gdy szlaban jest
zamknięty
const int OTWARTY = 90;    // Kat serwa, gdy szlaban jest
otwarty

Servo szlaban;

void setup() {
  pinMode(PRZYCISK_PIN, INPUT_PULLUP);
```

```
szlaban.attach(SERVO_PIN);
szlaban.write(ZAMKNIĘTY); // Zaczynam z zamkniętym szlabanem
}

void loop() {
  if (digitalRead(PRZYCISK_PIN) == LOW) { // Przycisk naciśnięty
    // Powoli obracaj w górę, stopień po stopniu
    for (int kat = ZAMKNIĘTY; kat <= OTWARTY; kat++) {
      szlaban.write(kat);
      delay(15);
    }
    delay(3000); // 3 sekundy otwarte:
    auto przejeżdża
    // Powoli znów zamykaj
    for (int kat = OTWARTY; kat >= ZAMKNIĘTY; kat--) {
      szlaban.write(kat);
      delay(15);
    }
  }
}
```

Co robi kod?

- Używamy biblioteki `Servo`. Dzięki `write()` mówisz, pod jakim kątem ma ustawić się serwo.
- Po naciśnięciu przycisku obracamy serwo stopień po stopniu od 0 do 90. Dzięki temu idzie spokojnie.
- Po 3 sekundach wracamy do 0: szlaban jest zamknięty.

Spróbuj...

- Niech szlaban otwiera się i zamyka szybciej
- Niech brzęczyk piszczy, gdy szlaban się porusza
- Niech czerwona LED świeci, gdy szlaban jest zamknięty, a zielona, gdy jest otwarty

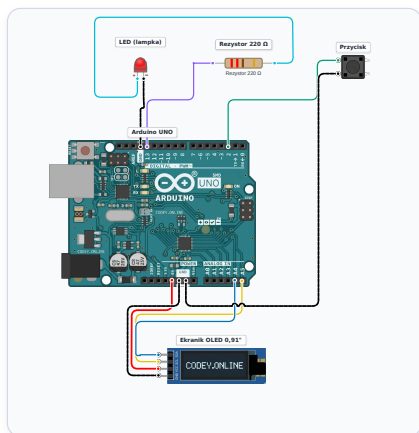
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/barrier i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

☆☆ Początkujący plus ⌚ 25 minut

Gra na refleks

Jak szybki jesteś? Poczekaj, aż LED się zaświeci. Potem jak najszybciej naciśnij przycisk. Ekran pokaże twój czas w milisekundach. Rzuć wyzwanie znajomym!

Schemat podłączenia



Czego potrzebujesz?



Tak to zbudujesz

1. Włóż przycisk nad rowkiem breadboardu: jedna nóżka do pin 2, nóżka po drugiej stronie do GND.
2. Podłącz LED z rezystorem 220 Ω do pin 13, skrótsz nóżkę do GND.
3. Podłącz ekran OLED: SDA do A4, SCL do A5, VCC do 5V i GND do GND.

Kod

```
// Gra na refleks
// Poczekaj, aż LED się włączy, i naciśnij przycisk jak najszybciej.
// Ekran pokaże, jak szybki byłeś!

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int PRZYCISK_PIN = 2; // Przycisk na pin 2 (druga strona do GND)
const int LED_PIN = 13; // LED na pin 13 (przez rezystor)

Adafruit_SSD1306 ekran(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

unsigned long najlepszyCzas = 0; // Twój najszybszy czas do tej pory

// Pokaż duży tekst na ekranie
void pokaz(const __FlashStringHelper* gora, unsigned long liczba) {
  ekran.clearDisplay();
  ekran.setTextSize(1);
  ekran.setCursor(0, 0);
  ekran.print(gora);
  if (liczba > 0) {
    ekran.setTextSize(2);
    ekran.setCursor(0, 14);
    ekran.print(liczba);
    ekran.print(F(" ms"));
  }
  ekran.display();
}

void setup() {
  pinMode(PRZYCISK_PIN, INPUT_PULLUP);
  pinMode(LED_PIN, OUTPUT);
  randomSeed(analogRead(A0)); // Daje naprawdę losowe czasy czekania
  ekran.begin(SSD1306_SWITCHCAPVCC, 0x3C);

  ekran.setTextColor(SSD1306_WHITE);
}

void loop() {
  pokaz(F("Naciśnij przycisk"), 0);
  while (digitalRead(PRZYCISK_PIN) == HIGH); // Czekamy, aż naciśniesz przycisk
  while (digitalRead(PRZYCISK_PIN) == LOW); // ...i znowu go puścisz
  pokaz(F("Gotow? Patrz na LED!"), 0);

  delay(random(2000, 5000)); // Tajny czas czekania od 2 do 5 sekund

  if (digitalRead(PRZYCISK_PIN) == LOW) { // Za wcześnie naciśnięte? Oszust!
    pokaz(F("Za szybko! Jeszcze"), 0);
    delay(2000);
    return;
  }

  digitalWrite(LED_PIN, HIGH); // TERAZ!
  unsigned long start = millis();
  while (digitalRead(PRZYCISK_PIN) == HIGH); // Czekamy na twoje naciśnięcie
  unsigned long czas = millis() - start;
  digitalWrite(LED_PIN, LOW);

  if (najlepszyCzas == 0 || czas < najlepszyCzas) {
    najlepszyCzas = czas;
    pokaz(F("NOWY REKORD!"), czas);
  } else {
    pokaz(F("Twój czas:"), czas);
  }
  delay(3000);
}
```

Co robi kod?

- Po naciśnięciu przycisku czekamy tajny czas od 2 do 5 sekund (`random()`).
- Potem LED się włącza i startujemy stoper z `millis()`.
- Gdy naciśniesz, liczymy, ile milisekund to trwało. Zapamiętujemy twój najlepszy czas.

Spróbuj...

- Zrób z tego grę dla dwóch graczy, każdy z własnym przyciskiem
- Niech buzzer zapiszczy przy nowym rekordzie
- Zagraj 5 rund, a potem pokaż swój średni czas

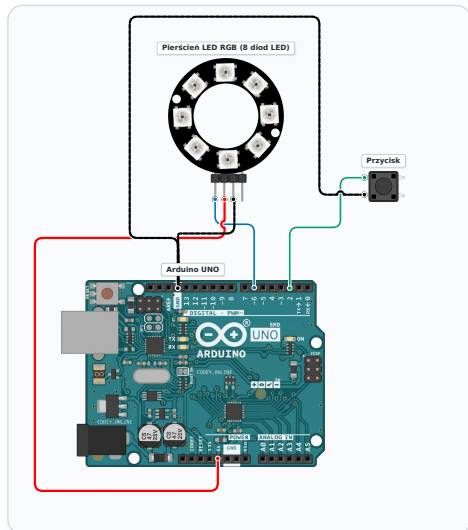
🔧 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/reaction-game i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★★ Początkujący plus ⏱ 15 minut

Elektroniczna kostka

Zrobisz kostkę ze światła. Naciśnij przycisk: lampki przez chwilę się toczą, a potem widzisz, ile wyrzuciłeś. Od 1 do 6 lampek!

Schemat podłączenia



Czego potrzebujesz?



1x Arduino UNO



1x Przycisk



1x Pierścień LED RGB (8 diod LED)



Kabel USB + Przewody połączeniowe

Tak to zbudujesz

- Włóż przycisk nad przerwą w breadboardzie: jedna nóżka do pin 2, a nóżka po drugiej stronie do GND.
- Podłącz LED-ring: DI do pin 6, 5V do 5V i GND do GND.

Kod

```
// Elektroniczna kostka
// Naciśnij przycisk: LED-ring chwilę się toczy, a potem świeci 1 do 6 lampek.

#include <Adafruit_NeoPixel.h>

const int PRZYCISK_PIN = 2;    // Przycisk na pin 2 (druga strona do GND)
const int PIERSCIEN_PIN = 6;   // DI LED-ring na pin 6
const int LICZBA_LED = 8;

Adafruit_NeoPixel pierscien(LICZBA_LED, PIERSCIEN_PIN, NEO_GRB + NEO_KHZ800);

// Zapala kilka lampek w jednym kolorze
void pokazLiczbe(int liczba, uint32_t kolor) {
  pierscien.clear();
  for (int i = 0; i < liczba; i++) {
    pierscien.setPixelColor(i, kolor);
  }
  pierscien.show();
}
```

```
void setup() {
  pinMode(PRZYCISK_PIN, INPUT_PULLUP);
  randomSeed(analogRead(A0)); // Sprawia, że kostka jest naprawdę losowa
  pierscien.begin();
  pierscien.setBrightness(50);
  pierscien.show();
}

void loop() {
  if (digitalRead(PRZYCISK_PIN) == LOW) {
    // Chwila "toczenia": szybko zmienne liczby, za każdym razem trochę wolniej
    for (int i = 0; i < 15; i++) {
      pokazLiczbe(random(1, 7), pierscien.Color(0, 80, 255));
      delay(40 + i * 10);
    }
    int rzut = random(1, 7); // Liczba od 1 do 6
    pokazLiczbe(rzut, pierscien.Color(0, 255, 60)); // Prawdziwa liczba na zielono
    delay(500);
  }
}
```

Co robi kod?

- `random(1, 7)` wybiera losową liczbę od 1 do 6.
- Najpierw pokazujemy szybko zmieniające się liczby na niebiesko. Wygląda to tak, jakby kostka się toczyła.
- Potem `pokazLiczbe()` pokazuje prawdziwą liczbę na zielono.

Spróbuj...

- Daj każdej liczbie własny kolor
- Niech brzęczyk krótko zapiszczy, gdy kostka się zatrzyma
- Zrób z tego kostkę od 1 do 8

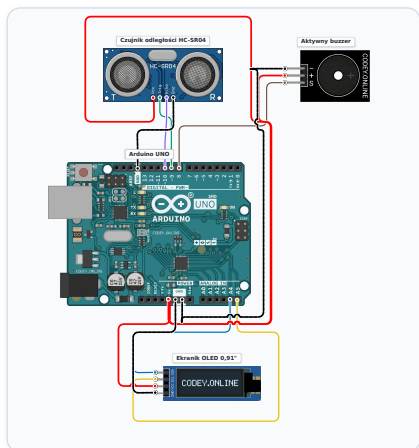
🚀 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/dice i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★★★★ Wyzwanie ⌚ 30 minut

Czujnik parkowania deluxe

To luksusowa wersja czujnika parkowania! Pika szybciej, gdy jesteś bliżej. A ekran OLED pokazuje odległość w centymetrach. Jest też pasek. Robi się dłuższy, gdy jesteś bliżej.

Schemat podłączenia



Czego potrzebujesz?



Tak to zbudujesz

1. Podłącz HC-SR04: Trig do pin 9, Echo do pin 10, Vcc do 5V i Gnd do GND.
2. Podłącz brzęczyk: S do pin 8, + do 5V i - do GND.
3. Podłącz ekran OLED: SDA do A4, SCL do A5, VCC do 5V i GND do GND.
4. Trzy części potrzebują 5V i GND. Użyj pasków + i - na swoim breadboard.

Kod

```
// Czujnik parkowania deluxe
// Jak zwykły czujnik parkowania, ale teraz widzisz odległość też na ekranie OLED.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int TRIG_PIN = 9; // Trig z HC-SR04 na pin 9
const int ECHO_PIN = 10; // Echo z HC-SR04 na pin 10
const int BRZECZYK_PIN = 8; // S brzęczyka na pin 8

Adafruit_SSD1306 ekran(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

// Mierzy odległość w centymetrach
long zmierzOdleglosc() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long czas = pulseIn(ECHO_PIN, HIGH, 30000);
  if (czas == 0) return 999; // Brak echa: nic blisko
  return czas / 58;
}

// Pokazuje dużą odległość na ekranie, z paskiem pod spodem
void pokazOdleglosc(long odleglosc) {
  ekran.clearDisplay();
  ekran.setTextSize(2);
  ekran.setCursor(0, 0);
  if (odleglosc >= 999) {
    ekran.print(F("----"));
  } else {
    ekran.print(odleglosc);
    ekran.print(F(" cm"));
  }
  // Pasek: im bliżej, tym dłuższy
  int dlugosc = map(constrain(odleglosc, 0, 100), 0, 100, 128, 0);
  ekran.fillRect(0, 24, dlugosc, 8, SSD1306_WHITE);
  ekran.display();
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BRZECZYK_PIN, OUTPUT);
  ekran.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  ekran.setTextColor(SSD1306_WHITE);
}

void loop() {
  long odleglosc = zmierzOdleglosc();
  pokazOdleglosc(odleglosc);

  if (odleglosc < 5) {
    digitalWrite(BRZECZYK_PIN, HIGH); // Bardzo blisko: jeden długi dźwięk
    delay(100);
  } else if (odleglosc < 50) {
    digitalWrite(BRZECZYK_PIN, HIGH); // Krótkie piknięcie
    delay(50);
    digitalWrite(BRZECZYK_PIN, LOW);
    delay(odleglosc * 10); // Bliżej = szybsze pikanie
  } else {
    digitalWrite(BRZECZYK_PIN, LOW);
    delay(100);
  }
}
```

Co robi kod?

- `zmierzOdleglosc()` mierzy odległość w centymetrach za pomocą HC-SR04.
- `pokazOdleglosc()` pokazuje dużą liczbę na ekranie i rysuje pasek przez `fillRect()`.
- Brzęczyk pika szybciej, gdy odległość jest mniejsza.

Spróbuj...

- Pokaż "STOP!" na ekranie, gdy jesteś bliżej niż 5 cm
- Pokaż odległość w metrach, gdy coś jest dalej niż 1 meter
- Dodaj pierścien LED, który zmienia kolor z zielonego na czerwony

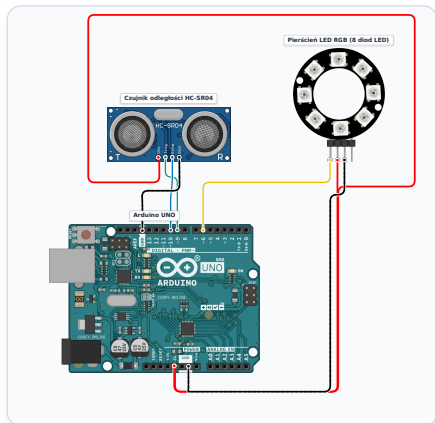
🚀 Zbuduj ten projekt z Codey: codey.online/pl/starter-kits/uno/projects/parking-sensor-deluxe i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★★★★ Wyzwanie ⌚ 25 minut

Miernik odległości z paskiem świetlnym

Zrobisz pasek świetlny, który pokazuje, jak blisko jest coś. Daleko: nie świeci żadna lampka. Im bliżej, tym więcej lampek się zapala. Najpierw zielone, potem pomarańczowe, a bardzo blisko czerwone.

Schemat podłączenia



Czego potrzebujesz?



1x Arduino UNO



1x Czujnik odległości HC-SR04



1x Pierścień LED RGB (8 diod LED)



Kabel USB + Przewody połączeniowe

Tak to zbudujesz

1. Podłącz HC-SR04: Trig do pin 9, Echo do pin 10, Vcc do 5V i Gnd do GND.
2. Podłącz pierścień LED: DI do pin 6, 5V do 5V i GND do GND.

Kod

```
// Miernik odległości z paskiem świetlnym
// Im bliżej jesteś, tym więcej lampek świeci: zielone, potem
// pomarańczowe, potem czerwone.

#include <Adafruit_NeoPixel.h>

const int TRIG_PIN = 9; // Trig z HC-SR04 na pin 9
const int ECHO_PIN = 10; // Echo z HC-SR04 na pin 10
const int RING_PIN = 6; // DI z pierścienia LED na pin 6
const int LICZBA_LED = 8;
const int MAX_ODLEGLOSC = 80; // Dalej niż to (w cm): wszystkie lampki
// zgaszone

Adafruit_NeoPixel pierscien(LICZBA_LED, RING_PIN, NEO_GRB + NEO_KHZ800);

long zmierzOdleglosc() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long czas = pulseIn(ECHO_PIN, HIGH, 30000);
  if (czas == 0) return 999;
  return czas / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pierscien.begin();

  pierscien.setBrightness(40);
  pierscien.show();
}

void loop() {
  long odleglosc = zmierzOdleglosc();
  Serial.print(F("Odleglosc: "));
  Serial.print(odleglosc);
  Serial.println(F(" cm"));

  // Ile lampek ma świecić? Blisko = 8, daleko = 0
  int liczba = map(constrain(odleglosc, 0, MAX_ODLEGLOSC), 0,
    MAX_ODLEGLOSC, LICZBA_LED, 0);

  for (int i = 0; i < LICZBA_LED; i++) {
    if (i < liczba) {
      if (i < 3) pierscien.setPixelColor(i, pierscien.Color(0, 255,
        0)); // Zielony
      else if (i < 6) pierscien.setPixelColor(i, pierscien.Color(255, 120,
        0)); // Pomarańczowy
      else pierscien.setPixelColor(i, pierscien.Color(255, 0,
        0)); // Czerwony
    } else {
      pierscien.setPixelColor(i, 0);
    }
  }
  // Wyłączony
  pierscien.show();
  delay(100);
}
```

Co robi kod?

- `zmierzOdleglosc()` mierzy odległość za pomocą HC-SR04.
- `map()` przelicza odległość na liczbę lampek: od 0 do 8.
- Pierwsze 3 lampki są zielone, następne 3 pomarańczowe, a ostatnie 2 czerwone.

Spróbuj...

- Spraw, żeby wszystkie lampki migały na czerwono, gdy jesteś bliżej niż 10 cm
- Dodaj brzęczyk, który piszczy przy czerwonym
- Zrób czulszy miernik: wszystkie lampki włączone przy 40 cm zamiast 80 cm

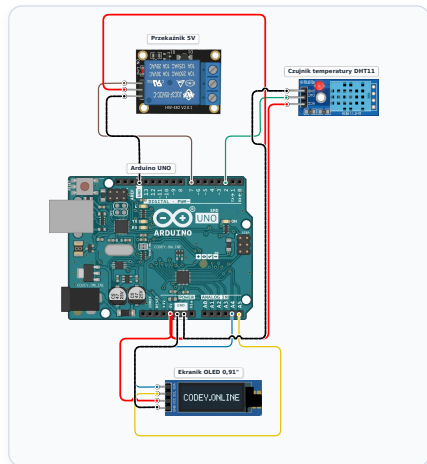
 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/distance-light-bar i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★★★★ Wyzwanie ⌚ 30 minut

Sprytny wentylator

Budujesz termostat. Gdy zrobi się cieplej niż 25 stopni, przekaźnik kliknie i się włączy. Tak Arduino może na przykład włączyć mały wentylator na baterię. Na ekranie widzisz temperaturę i czy wentylator jest włączony.

Schemat podłączenia



Czego potrzebujesz?



Tak to zbudujesz

1. Podłącz DHT11: DAT do pin 2, VCC do 5V i GND do GND.
2. Podłącz przekaźnik: S do pin 7, + do 5V i – do GND.
3. Podłącz ekran OLED: SDA do A4, SCL do A5, VCC do 5V i GND do GND.
4. Chcesz naprawdę coś przełączać? Użyj tylko baterii albo niskiego napięcia na COM i NO, nigdy gniazdka.

Kod

```
// Sprytny wentylator (termostat)
// Robi się za ciepło? Wtedy przekaźnik się włącza, na przykład dla
wentylatora.
// Na ekranie widzisz temperaturę i czy wentylator jest włączony.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2;      // DAT z DHT11 na pin 2
const int PRZEKAZNIK_PIN = 7; // S z przekaźnika na pin 7
const float WLACZ_POWYZEJ = 25.0; // Powyżej tej temperatury wentylator się
włącza
const float WYLACZ_PONIZEJ = 24.0; // Poniżej tej temperatury znów się
wyłącza

DHT dht(DHT_PIN, DHT11);
Adafruit_SSD1306 ekran(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

bool wentylatorWlaczony = false;

void setup() {
  Serial.begin(9600);
  pinMode(PRZEKAZNIK_PIN, OUTPUT);
  digitalWrite(PRZEKAZNIK_PIN, LOW); // Zaczniemy z wyłączonym przekaźnikiem
  dht.begin();
  ekran.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  ekran.setTextColor(SSD1306_WHITE);
}

void loop() {
  float temperatura = dht.readTemperature();

  if (!isnan(temperatura)) {
    // Dwie granice, aby przekaźnik nie klikał ciągle włącz i wyłącz
    if (temperatura > WLACZ_POWYZEJ) wentylatorWlaczony = true;
    if (temperatura < WYLACZ_PONIZEJ) wentylatorWlaczony = false;
    digitalWrite(PRZEKAZNIK_PIN, wentylatorWlaczony ? HIGH : LOW);
  }

  ekran.clearDisplay();
  ekran.setTextSize(2);
  ekran.setCursor(0, 0);
  if (isnan(temperatura)) {
    ekran.print(F("DHT11 ?"));
  } else {
    ekran.print(temperatura, 1);
    ekran.print((char)247);
    ekran.print(F("C"));
  }
  ekran.setTextSize(1);
  ekran.setCursor(0, 22);
  ekran.print(wentylatorWlaczony ? F("Wentyl.: WL") : F("Wentyl.: wyl"));
  ekran.display();

  Serial.print(F("Temp.: "));
  Serial.print(temperatura);
  Serial.println(wentylatorWlaczony ? F(" C -> WL") : F(" C -> wyl"));

  delay(2000);
}
```

Co robi kod?

- DHT11 mierzy temperaturę. Powyżej 25 stopni włączamy przekaźnik.
- Dopiero poniżej 24 stopni przekaźnik znowu się wyłączy. Dzięki tym dwóm granicom nie klika ciągle włącz i wyłącz.
- Ekran OLED pokazuje temperaturę oraz „Wentyl.: WL” albo „wyl”.

Spróbuj...

- Dodaj dwa przyciski, aby ustawiać temperaturę graniczną wyżej i niżej
- Niech LED świeci, gdy wentylator jest włączony
- Niech wentylator włączy się też wtedy, gdy wilgotność powietrza będzie powyżej 70%

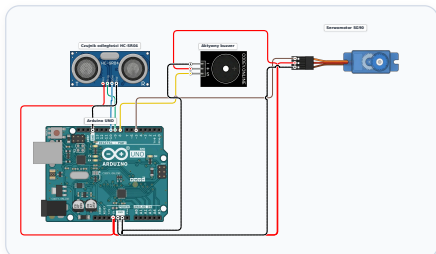
🔧 Zbuduj ten projekt z Codey: codey.online/pl/starter-kits/uno/projects/thermostat i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.

★★★★ Wyzwanie ⌚ 30 minut

Automatyczny szlaban

To sprytna wersja szlabanu! Czujnik odległości widzi, że zbliża się auto. Wtedy brzęczyk piszczy, a szlaban sam się otwiera. Auto odjechało? Wtedy znów się zamyka.

Schemat podłączenia



Czego potrzebujesz?



1x Arduino UNO



1x Czujnik odległości HC-SR04



1x Serwomotor SG90



1x Aktywny buzzer



Kabel USB + Przewody połączeniowe

Tak to zbudujesz

1. Podłącz HC-SR04: Trig do pinu 9, Echo do pinu 10, Vcc do 5V i Gnd do GND.
2. Podłącz servo: pomarańczowy do pinu 5, czerwony do 5V i brązowy do GND.
3. Podłącz brzęczyk: S do pinu 8, + do 5V i – do GND.
4. Przyklej słomkę albo patyczek po lodach do servo jako szlaban i ustaw czujnik przed nim.

Kod

```
// Automatyczny szlaban
// Czy jedzie auto (coś bliżej niż 15 cm)? Wtedy brzęczyk piszczy i
// szlaban się otwiera.

#include <Servo.h>

const int TRIG_PIN = 9;      // Trig z HC-SR04 na pinie 9
const int ECHO_PIN = 10;     // Echo z HC-SR04 na pinie 10
const int SERVO_PIN = 5;     // Pomarańczowy/żółty przewód servo na
// pinie 5
const int BRZECZYK_PIN = 8;  // S brzęczyka na pinie 8
const int BLISKI = 15;       // Od tej odległości (cm) szlaban się
// otwiera

const int ZAMKNIETE = 0;     // Kąt, gdy szlaban jest zamknięty
const int OTWARTE = 90;      // Kąt, gdy szlaban jest otwarty

Servo szlaban;

long zmierzOdleglosc() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long czas = pulseIn(ECHO_PIN, HIGH, 30000);
  if (czas == 0) return 999;
  return czas / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BRZECZYK_PIN, OUTPUT);
  szlaban.attach(SERVO_PIN);
  szlaban.write(ZAMKNIETE);
}

void loop() {
  long odleglosc = zmierzOdleglosc();

  if (odleglosc < BLISKI) {
    Serial.println(F("Auto! Szlaban otw."));
    digitalWrite(BRZECZYK_PIN, HIGH); // Krótki pisk
    delay(100);
    digitalWrite(BRZECZYK_PIN, LOW);

    szlaban.write(OTWARTE);
    delay(1000);
    // Zostaje otwarty, póki auto tam stoi
    while (zmierzOdleglosc() < BLISKI) {
      delay(200);
    }
    delay(2000); // Jeszcze chwilę czekamy, aż auto
    // naprawdę odjedzie
    szlaban.write(ZAMKNIETE);
    Serial.println(F("Szlaban zamk."));
  }
  delay(100);
}
```

Co robi kod?

- HC-SR04 cały czas mierzy odległość. Bliżej niż 15 cm? To znaczy, że stoi tam auto.
- Brzęczyk krótko piszczy, a servo obraca się na 90 stopni: otwarte.
- Dopóki auto tam stoi, szlaban zostaje otwarty. Potem czekamy 2 sekundy i szlaban się zamyka.

Spróbuj...

- Policz, ile aut przejechało, i pokaż to w monitorze szeregowym
- Dodaj czerwoną i zieloną LED jako światła przy szlabanie
- Niech szlaban otwiera się tylko wtedy, gdy naciśniesz też przycisk

 Zbuduj ten projekt z Codey: otwórz codey.online/pl/starter-kits/uno/projects/automatic-barrier i kliknij ZRÓB TO. Wtedy kod i schemat podłączenia będą od razu gotowe.