

Krok 3: Po zakończeniu oceny wykonaj odpowiednią procedurę, na przykład jeśli otrzymasz: Jeśli „numer karty identyfikacyjnej” jest taki sam jak „wartość klucza” przycisku „A”, będzie sterować światłem RGB aby włączyć czerwoną, jeśli jest taki sam jak „wartość klucza” przycisku „B”, będzie sterować światłem RGB, aby włączyć niebieski i tak dalej. Stałe porównuj i oszczędzaj, a następnie wykonuj odpowiednie instrukcje.

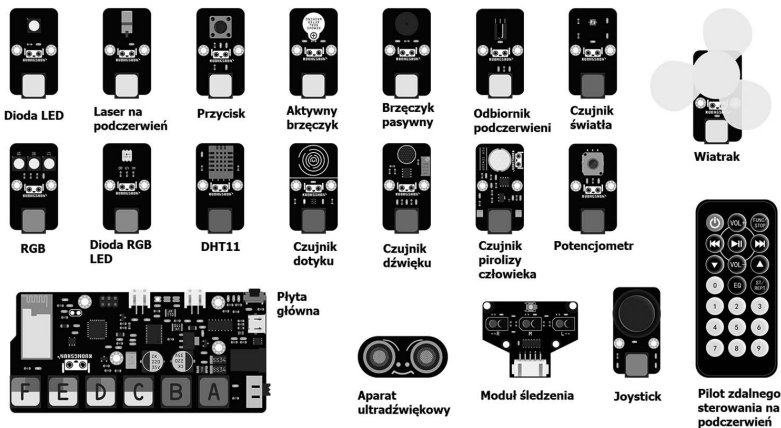
KOD:

```
#include <IRremote.h>
long ir_item;
IRrecv irrecv_B(8);
decode_results results_B;

void setup()
{
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  irrecv_B.enableIRIn();
}

void loop()
{
  if (irrecv_B.decode(&results_B))
  {
    ir_item=results_B.value;
    String type="UNKNOWN";
    String typeList["RCMUSEP",
"RC5", "RC6", "NEC", "SONY", "PANASONIC", "JVC", "SAMSUNG", "MITSUBISHI", "DAEWOO", "LG", "SANYO", "MITSUBISHI", "DISH", "SHARP", "DENON", "PRONTO", "LEGO_PF"];
    if(results_B.decode_type==1&&results_B.decode_type<17)
    {
      type=typeList[results_B.decode_type];
    }
    Serial.println("IR TYPE:"+type+"");
    Serial.println(ir_item,HEX);
    if (ir_item == 0x9035E63)
    {
      analogWrite(10,255);analogWrite(11,0);
      Write(9,0);analogWrite(11,0);
    }
    else if (ir_item == 0x70EDD433)
    {
      analogWrite(10,0);analogWrite(9,255);analogWrite(11,0);
    }
    else if (ir_item == 0x472581EC)
    {
      analogWrite(10,0);analogWrite(9,0);analogWrite(11,255);
    }
    else if (ir_item == 0x2AFBA97)
    {
      analogWrite(10,0);analogWrite(9,0);analogWrite(11,0);
      irrecv_B.resume();
    }
    else
    {
  }
  }
}
```

Instrukcja modułu



nova szkoła
ul. POW 25, 90-248 Łódź,
www.nowaszkoła.com
tel. (42) 630 17 28,
(42) 630 04 88, fax: (42) 632 73 28

OSTRZEŻENIA!



1. Produkt nie jest przeznaczony dla dzieci poniżej 3 lat. Zawiera małe elementy – ryzyko zadławienia.
2. Do użytku pod bezpośrednim nadzorem osoby dorosłej.
3. Należy zachować opakowanie lub/i instrukcję. Zawierają one ważne informacje mogące być przydatne w przyszłości.
4. Użytkowanie niezgodne z zaleceniami zwalnia producenta od odpowiedzialności za ewentualne szkody.



Robot Small Car. Zestaw konstrukcyjny VR 0012

Wiek

■ 8+

Waga

■ 1,2 kg



Prezentowany zestaw powstał z myślą o tych, którzy chcą zwiększyć swoją wiedzę w zakresie środowiska programistycznego Arduino oraz konstruowania robotów.

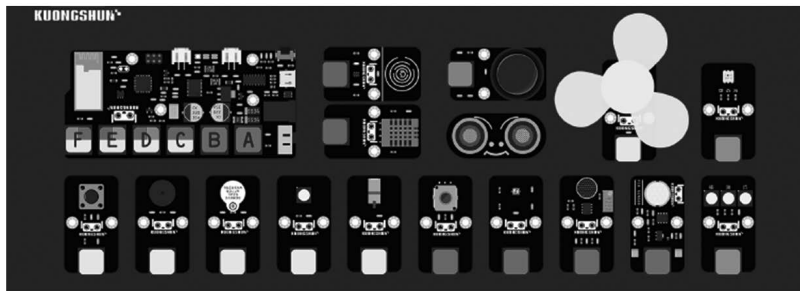
Dzięki jego strukturze zdobyć można zarówno wiedzę teoretyczną, jak i przydatną wiedzę praktyczną. Sprawdzisz w praktyce działanie czujników, sterowników silników oraz wielu innych elementów. Zdobyte umiejętności będziesz mógł wykorzystać do budowy wielozadaniowego robota, który będzie potrafił m.in. omijać przeszkody, szukać źródła najsilniejszego światła i wykonywać komendy wysłane do niego za pomocą pilota. Wykonując różne eksperymenty nauczysz się budować programowalne urządzenia. Zostań Mistrzem Robotyki! Poznaj podstawy programowania i zbuduj wielozadaniowego robota.

Specyfikacja produktu

zawartość zestawu:

- zestaw śrubek i nakrętek,
- elementy nadwozia,
- dwa koła,
- płyta Arduino UNO R69 RJ12,
- pilot,
- wiatrak,
- śrubokręt,
- blaszki zaciskowe,
- taśma izolacyjna 5mx18mm,
- serwo SG-90,
- kabel USB,
- 2x silnik napędowy do kół,
- ładowarka sieciowa,
- uchwyty montażowe,
- plastikowe trzecie koło,
- silikonowy przewód przedłużający do baterii,
- 5x przewód do programowania,
- kaseta na baterie,

- 2x moduł joystick,
- ultradźwiękowy miernik odległości,
- moduł podczerwieni,
- dioda LED,
- moduł dotykowy,
- port zasilający,
- czujnik temperatury,
- moduł brzęczyka,
- serwomotor DC,
- czujnik dźwięku,
- przycisk,
- czujnik światła,
- moduł sygnalizacja świetlna,
- laserowa dioda LED,
- moduł SMD RGB,
- potencjometr,
- czujnik pirolizy człowieka,
- plastikowa walizka.



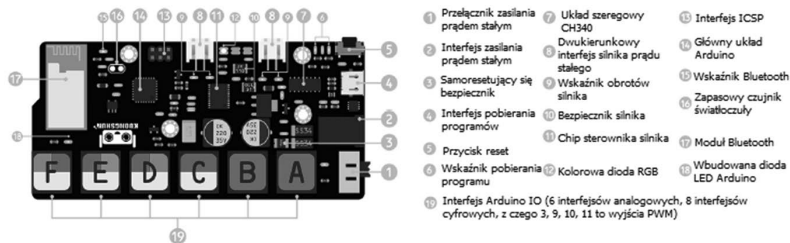
RJ25: Zestaw samochodowy z kryształową głowicą. Łatwy do podłączenia.

Płyta główna. Wprowadzenie

Aby nauka programowania była wygodniejsza i bezpieczniejsza dla dzieci, opracowaliśmy tę płytkę sterującą w oparciu o układ ATmega328 ArduinoUNO. Jego parametry są dokładnie takie same jak oficjalnego UNO, nie ma dużej różnicy w procesie użytkowania, a ponadto integruje moduł Bluetooth i napęd silnikowy. Bardzo łatwy w użyciu!

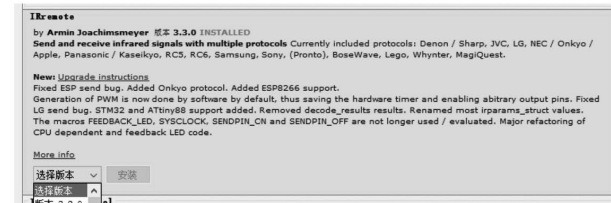
Płytkę jest przełomem łączącym piny z gniazdami kabla sieciowego z kryształową głowicą. Każde gniazdo odpowiada wielu pinom Arduino, w tym pinom cyfrowym, pinom analogowym i pinom PWM. Może zaspokoić większość potrzeb związanych z uczeniem się i rozwojem w środowisku programistycznym.

Poniżej znajduje się wprowadzenie do sekcji płyty.

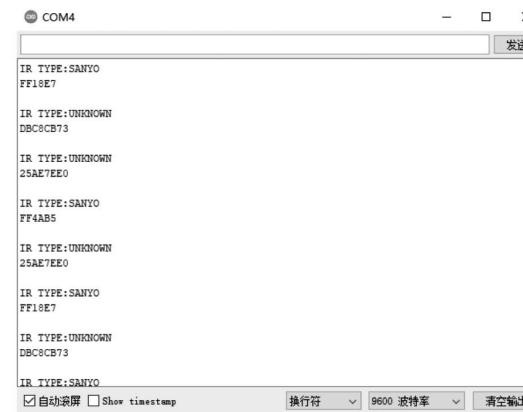
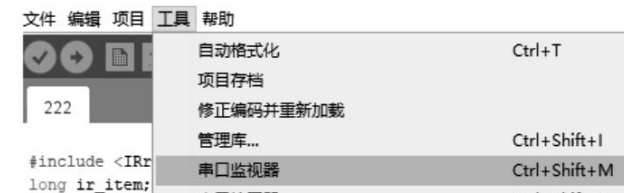


1. Przełącznik zasilania prądem stałym
 2. Interfejs zasilania prądem stałym
 3. Samoresełujący się bezpiecznik
 4. Interfejs pobierania programów
 5. Przycisk reset
 6. Wskaźnik pobierania programu
 7. Układ szeregowy CH340
 8. Dwukierunkowy interfejs silnika prądu stałego
 9. Wskaźnik obrotów silnika
 10. Bezpiecznik silnika
 11. Chip sterownika silnika
 12. Interfejs ICSP
 13. Główny układ Arduino
 14. Wskaźnik Bluetooth
 15. Zapasowy czujnik światłoczuły
 16. Moduł Bluetooth
 17. Wbudowana dioda LED Arduino
18. Interfejs Arduino IO (6 interfejsów analogowych, 8 interfejsów cyfrowych, z czego 3, 9, 10, 11 to wyjścia PWM)

Wybierz odpowiednią wersję „3.2.0” i kliknij „Zainstaluj”, aby zakończyć ładowanie biblioteki.



Wybierz „Narzędzia” na pasku menu i kliknij „Monitor szeregowy”.



Port szeregowy wyświetla odczytaną wartość klucza.

Pilot na podczerwień światła RGB

Po odczytaniu danych przycisku możemy sterować innymi modułami oceniając wartość przycisku. Zasada jest taka sama, jak w przypadku sterowania diodą LED wartością czujnika w poprzednim kursie.

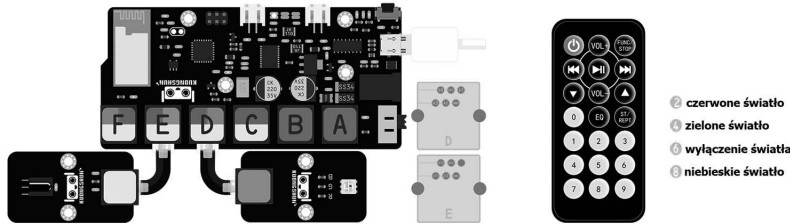
Krok 1: Odczytaj kluczową wartość pilota. Różne piloty zdalnego sterowania mają różne wartości klawiszy. Jaka jest więc kluczowa wartość? Możesz zrozumieć, że każdy ma unikalny numer identyfikacyjny.

Krok 2: Po naciśnięciu określonego klawisza pilot wyśle „numer karty identyfikacyjnej” tego klucza do urządzenia odbiorczego, a urządzenie odbiorcze oceni go po odebraniu „numeru karty identyfikacyjnej” (niezależnie od tego, czy jest taki sam jak dane przechowywane wewnętrznie).

Odczytywanie danych zdalnego sterowania

Nasz moduł odbiornika podczerwieni obsługuje inne piloty zdalnego sterowania, ale ponieważ kluczowa wartość każdego pilota jest inna, podczas korzystania z niego należy najpierw odczytać kluczową wartość pilota. Teraz najpierw nauczymy Cię, jak odczytać wartość klucza pilota.

Prześlij kod do Arduino, otwórz asystenta portu szeregowego i naciśnij dowolny przycisk na pilocie, aby uzyskać kluczową wartość bieżącego przycisku.



Kod odczytu wartości klucza pilota na podczerwień:

```
volatile char data;
void yf_setMotor(int motorId, int motorDir, int speed)
{
    int speedPin, directionPin;
    if (motorId == 1) // motor 1
    {
        speedPin = 5;
        directionPin = 4;
    }
    else if (motorId == 2) // motor 2
    {
        speedPin = 6;
        directionPin = 7;
    }
    else
    {
        return;
    }
    if (speed == 0)
    {
        digitalWrite(speedPin, 0);
    }
    else
    {
        if (motorDir == 0)
        {
            digitalWrite(directionPin, LOW);
            analogWrite(speedPin, speed);
        }
        else if (motorDir == 1)
        {
            digitalWrite(directionPin, HIGH);
            analogWrite(speedPin, speed);
        }
    }
}

void Forward()
{
    yf_setMotor(1,0,200);
    yf_setMotor(2,0,200);
}

void Back()
{
    yf_setMotor(1,1,180);
    yf_setMotor(2,1,180);
}

void Left()
{
    yf_setMotor(1,1,200);
    yf_setMotor(2,0,200);
}

void Right()
{
    yf_setMotor(1,0,200);
    yf_setMotor(2,1,200);
}

void Stop()
{
    yf_setMotor(2,0,0);
    yf_setMotor(1,0,0);
}

void setup()
{
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    Serial.begin(9600);
    data = 0;
}

void loop()
{
    while (Serial.available() > 0)
    {
        data = Serial.read();
        Serial.println(data);
    }
    if (data == 'f')
    {
        Forward();
    }
    else if (data == 'b')
    {
        Back();
    }
    else if (data == 'l')
    {
        Left();
    }
    else if (data == 'r')
    {
        Right();
    }
    else if (data == 's')
    {
        Stop();
    }
    else
    {
        Stop();
    }
}
}
```

Kiedy pojawi się następujący monit, musisz załadować odpowiedni plik biblioteki.



Pozwól, że nauczę Cię, jak ładować i dodawać biblioteki. Najpierw wybierz „Projekt” w menu powyżej w pierwszym kroku, wybierz „Załaduj bibliotekę” w drugim kroku i wybierz „Zarządzaj biblioteką” w trzecim kroku.



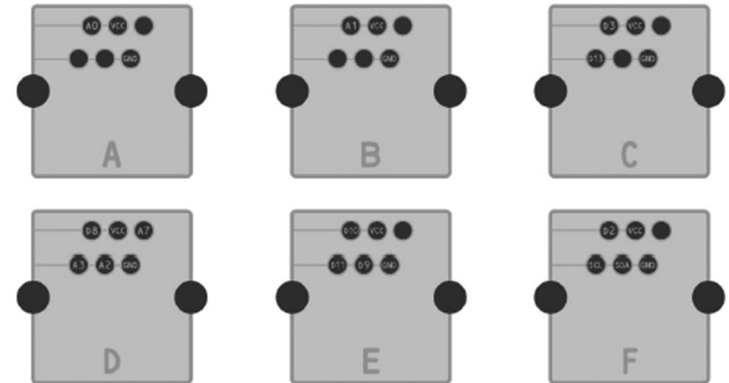
Parametry układu ATmega328 na płytce:

- Mikrokontroler / ATmega328.
- Napięcie robocze / 5v.
- Szybkość zegara / 16MHz.
- SRAM / 2 KB (ATmega328).
- Wejście analogowe / 6.
- EEPROM / 1 KB (ATmega328).
- Prąd na pin I/O: 40mA.
- Prąd pinu 3,3V / 50mA.
- EEPROM / 1 KB (ATmega328).
- Napięcie wejściowe / przy podłączeniu do USB lub zewnętrznego wejścia 7-12 V DC nie jest wymagane zewnętrzne zasilanie.
- Napięcie wyjściowe: wyjście 5 V DC i wyjście 3,3 V DC.
- Pin cyfrowy I/O: 14 (sześć z nich zapewnia wyjście PWM).
- Pamięć Flash / 32KB(ATmega328) program ładujący zajmuje 0,5KB.

Płyta główna zastosowana w naszym zestawie jest zmodyfikowana przez Arduino UNO, więc jej piny i funkcje są dokładnie takie same jak oryginalne UNO. Jednak w celu zmniejszenia uciążliwego okablowania w procesie uczenia się, zastąpiliśmy tradycyjny interfejs interfejsem kabla sieciowego RJ25 z kryształową głowicą, a jego porty i odpowiadające im piny są takie, jak pokazano w tabeli.

Wybór interfejsu	Odpowiedni pin	Opis
A	A0	Jeden niezależny interfejs
B	A1	Jeden niezależny interfejs
C	D3 (PWM) D13	Multipleksowanie wielu interfejsów
D	A2, A3, A7, D8	Multipleksowanie wielu interfejsów
E	D9, D10, D11	Pełny interfejs PWM
F	A4, A5, D2	Multipleksowanie wielu interfejsów

Opisy styków w tabeli można zrozumieć, odwołując się do rysunku „Arduino-Port” w informacjach pomocniczych.



Oprócz wyżej wymienionych pinów, jest kilka pinów, których używamy do rozbudowy innych modułów, takich jak napęd silnika, Bluetooth itp. Nie musisz tego rozumieć, po prostu używaj ich intuicyjnie.

Sekcja 1. Moduł LED

Przebieg

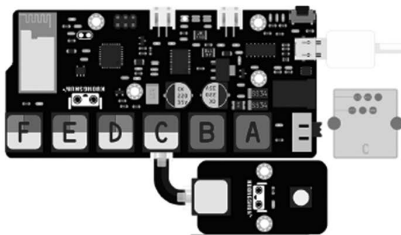
W tej lekcji nauczysz się, jak zmieniać jasność diody LED, używając różnych wartości rezystora.

LED

Diody LED są świetnymi wskaźnikami świetlnymi. Zużywają bardzo mało energii elektrycznej i są prawie wieczne. Nie można bezpośrednio podłączyć diody LED do akumulatora lub źródła napięcia, ponieważ:

1. dioda LED ma przewód dodatni i ujemny i nie zaświeci się, jeśli zostanie umieszczona w niewłaściwy sposób, oraz
2. dioda LED musi być używana z rezystorem w celu ograniczenia lub „dławienia” ilość przepływającego przez niego prądu, inaczej się wypali!

Jeśli nie użyjesz rezystora z diodą LED, może on prawie natychmiast ulec zniszczeniu, ponieważ przepłynie przez niego zbyt duży prąd, ogrzewając go i niszcząc „złącze”, w którym wytwarzane jest światło.



PIN:

PORT	PIN
C	D3

KOD:

```
void setup() {
  pinMode(D3, OUTPUT);
}
void loop() {
  digitalWrite(D3, HIGH);
  delay(1000);
  digitalWrite(D3, LOW);
  delay(1000);
}
```

Sekcja 2. Moduł sygnalizacji świetlnej

Przebieg

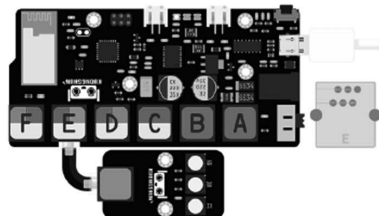
W tej lekcji dowiesz się, jak zmienić jasność diody LED za jej pomocą.

PIN:

PORT	PIN	Opis
E	D9	zielona dioda LED
	D10	żółta dioda LED
	D11	czerwona dioda LED

KOD:

```
void setup() {
  pinMode(9, OUTPUT); //green
  pinMode(10, OUTPUT); //yellow
  pinMode(11, OUTPUT); //red
}
void loop() {
  digitalWrite(9, HIGH);
  delay(10000);
  digitalWrite(9, LOW);
  digitalWrite(10, HIGH);
  delay(4000);
  digitalWrite(10, LOW);
  digitalWrite(11, HIGH);
  delay(5000);
}
```



KOD:

```
volatile char data;
void yf_setMotor(int motorId, int motorDir,
int speed)
{
  int speedPin, directionPin;
  if (motorId == 1) // motor 1
  {
    speedPin = 5;
    directionPin = 4;
  }
  else if (motorId == 2) // motor 2
  {
    speedPin = 6;
    directionPin = 7;
  }
  else
  {
    return;
  }
  if (speed == 0)
  {
    digitalWrite(speedPin, 0);
  }
  else
  {
    if(motorDir == 0)
    {
      digitalWrite(directionPin, LOW);
      analogWrite(speedPin, speed);
    }
    else if(motorDir == 1)
    {
      digitalWrite(directionPin, HIGH);
      analogWrite(speedPin, speed);
    }
  }
}
void Forward()
{
  yf_setMotor(1,0,200);
  yf_setMotor(2,0,200);
}
void Back()
{
  yf_setMotor(1,1,180);
  yf_setMotor(2,1,180);
}
void Left()
{
  yf_setMotor(1,1,200);
  yf_setMotor(2,0,200);
}
void Right()
{
  yf_setMotor(1,0,200);
  yf_setMotor(2,1,200);
}
void Stop()
{
  yf_setMotor(2,0,0);
  yf_setMotor(1,0,0);
}
void setup()
{
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  Serial.begin(9600);
  data = 0;
}
void loop(){
  while (Serial.available() > 0)
  {
    data = Serial.read();
    Serial.println(data);
  }
  if (data == 'f')
  {
    Forward();
  }
  else if (data == 'b')
  {
    Back();
  }
  else if (data == 'l')
  {
    Left();
  }
  else if (data == 'r')
  {
    Right();
  }
  else if (data == 's')
  {
    Stop();
  }
  else
  {
    Stop();
  }
}
```

Sekcja 18. Pilot zdalnego sterowania na podczerwień

Przebieg

W tej lekcji nauczysz się używać pilota na podczerwień do sterowania oświetleniem RGB.

Pilot na podczerwień

Obwód nadawczy pilota na podczerwień wykorzystuje diody emitujące światło podczerwone do emitowania modulowanych fal świetlnych w podczerwieni. Obwód odbiorczy podczerwieni składa się z diod odbiorczych podczerwieni, triod lub fotokomórek krzemowych, które przetwarzają światło podczerwone emitowane przez nadajnik podczerwieni na odpowiednie sygnały elektryczne, a następnie wysyłają je. Ustaw wzmacniacz.

Pilot na podczerwień to bezprzewodowa, bezdotykowa technologia sterowania, która ma znaczące zalety, takie jak silne zdolności przeciwwzakłóceńowe, niezawodna transmisja informacji, niskie zużycie energii, niski koszt i łatwa implementacja.

Myślę, że każdy jest zaznajomiony z pilotem na podczerwień, ponieważ jest to najczęściej używany kontroler w życiu. Wiele urządzeń elektrycznych w domu jest wyposażonych w pilota na podczerwień. Jest łatwy w użyciu i tani. Na tym kursie użyjemy pilota na podczerwień do zmiany koloru światła RGB i wyłączenia go.

PIN:

PORT	PIN
E	D10
D	D8

Opis interfejsu:

- 1: Instrukcja użytkownika: Ustal, jak ustawić kluczowe parametry i dodać interfejs sterowania.
- 2: Ikona stanu połączenia Bluetooth:
 - ✖ Bluetooth nie jest podłączony
 - ✔ Połączenie Bluetooth powiodło się
- 3: Dodaj klucz do strony kontrolnej.
- 4: Obszar przycisku DIY.

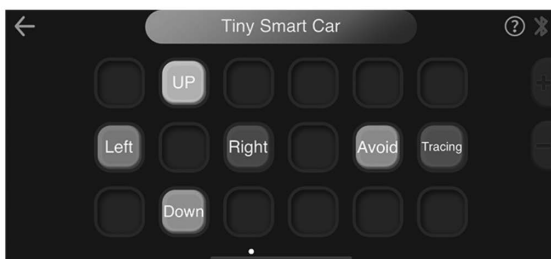
Połączenie Bluetooth

Naciśnij długo dowolny przycisk, aby wejść do ustawienia, wprowadź nazwę przycisku w ustawieniach, naciśnij klawisze (cyfry i litery, to polecenie zostanie wysłane do Arduino) i kolor, który lubisz, jak pokazano na rysunku 6.

Rysunek 6. Interfejs ustawień przycisków



Rysunek 7. Ustawienie przycisku zostało zakończone



Po zakończeniu ustawień podłącz Arduino do komputera, podłącz Bluetooth telefonu komórkowego do Arduino, otwórz asystenta portu szeregowego, naciśnij właśnie ustawiony przycisk, a zobaczysz odpowiednie polecenie w asystencie portu szeregowego. Rozpocznij teraz swoje szalone tworzenie i wypal swój mózg do syta. Nie ma nic, czego nie możesz zrobić, jest tylko to, o czym nie możesz pomyśleć.

W programie trzy diody świecą się cyklicznie po kolei, zielone światło to 10s, żółte światło to 4s, a czerwone światło to 5s.

Sekcja 3. Moduł brzęczyka

Przebieg

W tej lekcji nauczysz się zmieniać głośność brzęczyka za jego pomocą.

Brzęczyk

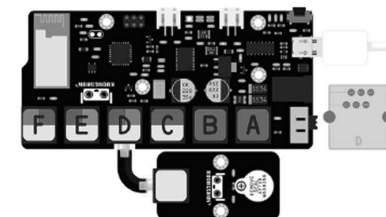
Elektroniczne brzęczyki są zasilane prądem stałym i wyposażone w układ scalony. Są szeroko stosowane w komputerach, drukarkach, kserokopiarkach, alarmach, zabawkach elektronicznych, samochodowych urządzeniach elektronicznych, telefonach, timerach i innych produktach elektronicznych do urządzeń głosowych. Brzęczyki można podzielić na aktywne i pasywne. Obróć styki dwóch brzęczyków do góry. Ten z zieloną płytką to brzęczyk pasywny, a drugi oklejony czarną taśmą to brzęczyk aktywny. Różnica między nimi polega na tym, że aktywny brzęczyk ma wbudowane oscylujące źródło, więc będzie generować dźwięk po naelektryzowaniu. Pasywny brzęczyk nie ma takiego źródła, więc nie będzie tweekował, jeśli używane są sygnały DC. Zamiast tego musisz użyć fal prostokątnych, których częstotliwość wynosi od 2K do 5K, aby go napędzać. Aktywny brzęczyk jest często droższy niż pasywny ze względu na wiele wbudowanych obwodów oscylacyjnych.

PIN:

PORT	PIN
D	D8

KOD:

```
void setup() {  
  pinMode(D8, OUTPUT);  
}  
void loop() {  
  digitalWrite(D8, HIGH);  
  delay(1000);  
  digitalWrite(D8, LOW);  
  delay(1000);  
}
```



Sekcja 4. Moduł lasera na podczerwień

Przebieg

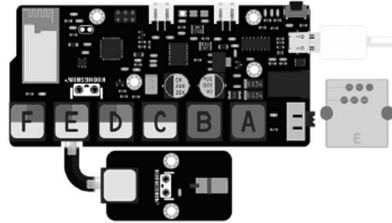
Zwykle lasery są takie same jak diody LED. Mogą emitować światło tylko wtedy, gdy są zasilane, ale w przeciwieństwie do diod LED lasery mają dobrą monochromatyczność, wysoką jasność i stabilną kierunkowość. W porównaniu z diodami LED mają szerszy zakres zastosowań. Na przykład nauczyciele używają ich w klasie. Wskaźniki laserowe, światłowody stosowane w komunikacji, poziomiczne elektroniczne stosowane w budynkach oraz popularne wózki laserowe... wszystkie te urządzenia w pełni wykorzystują głowice laserowe.

PIN:

PORT	PIN
E	D10

KOD:

```
void setup() {
  pinMode(D10, OUTPUT);
}
void loop() {
  digitalWrite(D10, HIGH);
  delay(1000);
  digitalWrite(D10, LOW);
  delay(1000);
}
```



Sekcja 5. Moduł przycisków

Przebieg

W tej lekcji nauczysz się używać przycisków z wejściami cyfrowymi do włączania i wyłączania diody LED. Naciśnięcie przycisku włączy diodę LED. Naciśnięcie drugiego przycisku wyłączy diodę LED.

Przełączniki wciskane

Przełączniki to naprawdę proste elementy. Kiedy naciśniesz przycisk lub przesuniesz dźwignię, łączą się ze sobą dwa styki, dzięki czemu może przez nie przepływać prąd.

Małe przełączniki dotykowe używane w tej lekcji mają cztery połączenia, co może być nieco mylące.



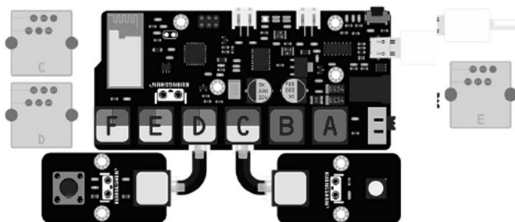
Właściwie są tylko dwa połączenia elektryczne. Wewnątrz zespołu przełączników styki B i C są ze sobą połączone, podobnie jak A i D.

PIN:

PORT	PIN
D	D8

KOD:

```
void setup() {
  pinMode(8, INPUT);//green
  pinMode(3, OUTPUT);//yellow
}
void loop() {
  if (digitalRead(8) {
    digitalWrite(3, HIGH);
  } else {
    digitalWrite(3, LOW);
  }
}
```



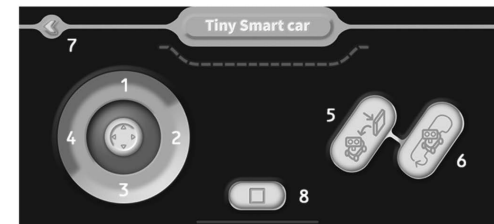
Rysunek 2. Interfejs potwierdzenia połączenia Bluetooth



Rysunek 3. Interfejs połączenia Bluetooth



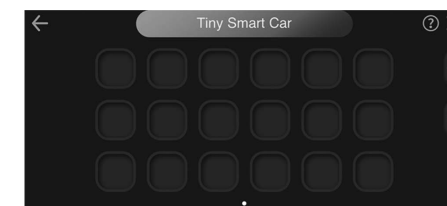
Rysunek 4. Interfejs sterowania joystickiem



1. Kierunek pojazdu.
2. Samochód skręca w prawo.
3. Tylony kierunek pojazdu.
4. Kierunek skrętu w lewo samochodu.
5. Jednym kliknięciem przełącz na tryb bariery ultradźwiękowej.
6. Jeden klucz przełącza w tryb śledzenia.
7. Powrót do poprzedniego menu.
8. Przycisk zatrzymania samochodu.

4.2. Kontrola DIY

Kliknij ikonę sterowania DIY na rysunku 1, aby przejść do interfejsu sterowania (rysunek 5). W interfejsie sterowania DIY możesz określić kolor przycisku, funkcję przycisku itp.



Rysunek 5. Interfejs sterowania DIY



Interfejs wyboru pojazdu

4. Instrukcje wyboru trybu i sterowania

Jesteśmy wyposażeni w dwa tryby dla każdego samochodu, jeden rodzaj to „sterowanie joystickiem”. Służy to bezpośrednio do sterowania ruchem samochodu. Odpowiednie znaki zostały zapisane w aplikacji, a samochód można przesunąć, przeciągając suwak. Drugi to „moduł DIY”. Ten tryb wymaga samodzielnego ustawienia postaci, co może wydawać się bardziej kłopotliwe, ale jego grywalność i możliwości majsterkowania są potężniejsze, takie jak możliwość kontrolowania włączania i wyłączenia diody LED oraz otwierania i zamykania kontroli dostępu.



Rysunek 1. Interfejs wyboru trybu pojazdu

4.1. Sterowanie joystickiem

Kliknij ikonę, aby wejść do interfejsu sterowania, następnie aplikacja wyświetli okno, jak pokazano na rysunku 2 z pytaniem, czy chcemy podłączyć urządzenie, klikamy przycisk „OK”, aplikacja automatycznie połączy się z samochodem (rysunek 3). Po nawiązaniu połączenia wejdź do interfejsu sterowania (jak pokazano na rysunku 4), przesunij moduł i samochód może się poruszać.

Aby użyć przycisku do sterowania diodą LED, najpierw odczytaj stan przycisku, następnie oceń stan przycisku, a na koniec steruj diodą LED zgodnie ze stanem przycisku.

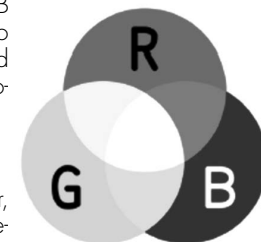
Sekcja. 6 Moduł RGB-LED

Przebieg

Omówienie diody LED RGB to zabawny i łatwy sposób na dodanie koloru do projektów. Są jak 3 zwykłe diody LED w jednej.

RGB

Na pierwszy rzut oka diody LED RGB (czerwony, zielony i niebieski) wyglądają jak zwykłe diody LED. Jednak w zwykłym pakiecie LED znajdują się w rzeczywistości trzy diody LED, jedna czerwona, jedna zielona i jedna niebieska. Kontrolując jasność każdej z poszczególnych diod LED, możesz mieszać barwy i uzyskać prawie każdy kolor, jaki chcesz. Mieszymy kolory w taki sam sposób, jak mieszymy farby na palecie – dostosowując jasność każdej z trzech diod LED. Trudnym sposobem na to byłoby użycie rezystorów o różnych wartościach (lub rezystorów zmiennych), dałoby to dużo pracy! Na szczęście dla nas płyta UNO R3 posiada funkcję analogWrite, której można używać z pinami oznaczonymi ~, aby wyprawdzać zmienną moc do odpowiednich diod LED. Dioda LED RGB ma cztery przewody. Jeden przewód prowadzi do dodatniego połączenia każdej z pojedynczych diod LED w pakiecie, a pojedynczy przewód jest podłączony do wszystkich trzech ujemnych stron diod LED.



Kolor

Powodem, dla którego możesz mieszać dowolny kolor, zmieniając ilość światła czerwonego, zielonego i niebieskiego, jest to, że twoje oko ma trzy typy receptorów światła (czerwony, zielony i niebieski). Twoje oko i mózg przetwarzają ilości czerwieni, zieleni oraz błękitu i przekształcają je w kolor widma.

W pewnym sensie, używając trzech diod, płatamy figla oku. Ten sam pomysł jest stosowany w telewizorach, w których wyświetlacz LCD ma czerwone, zielone i niebieskie kropki obok siebie, tworzące każdy piksel.

Jeśli ustawimy jasność wszystkich trzech diod na taką samą, to ogólny kolor światła będzie biały. Jeśli wyłączymy niebieską diodę LED, tak aby tylko czerwona i zielona dioda LED miały tę samą jasność, wtedy światło będzie żółte. Możemy oddzielnie sterować jasnością każdej z części czerwonej, zielonej i niebieskiej diody LED, co umożliwi dowolne mieszanie kolorów.

Czerń to nie tyle kolor, co brak światła. Dlatego najbliższą czerni z naszą diodą LED jest wyłączenie wszystkich trzech kolorów.

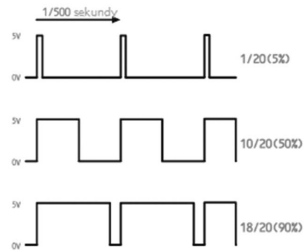
Teoria (PWM)

Modulacja szerokości impulsu (PWM) to technika kontrolowania mocy. Używamy go również tutaj do sterowania jasnością każdej z diod LED. Poniższy schemat przedstawia sygnał z jednego z pinów PWM na UNO.

Mniej więcej co 1/500 sekundy wyjście PWM generuje impuls. Długość tego impulsu jest kontrolowana przez funkcję 'analogWrite'. Tak więc 'analogWrite(0)' w ogóle nie wytworzy żadnego impulsu, a 'analogWrite(255)' wytworzy impuls, który będzie trwał aż do następnego impulsu, tak że wyjście jest faktycznie włączone przez cały czas.

Jeśli określimy wartość w analogWrite, która jest gdzieś pomiędzy 0 a 255, wtedy wytworzymy impuls. Jeśli impuls wyjściowy jest wysoki tylko przez 5% czasu, to czymkolwiek wyjdziemy, otrzymamy tylko 5% pełnej mocy.

Jeśli jednak napięcie wyjściowe wynosi 5 V przez 90% czasu, obciążenie otrzyma 90% dostarczanej do niego mocy. Nie widzimy, jak diody LED włączają się i wyłączają z taką prędkością, więc dla nas wygląda to tak, jakby jasność się zmieniała.

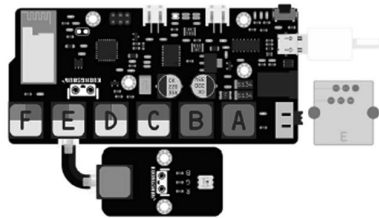


PIN:

PORT	PIN	Opis
E	D9	zielony
	D10	czerwony
	D11	niebieski

KOD:

```
#define BLUE 11
#define GREEN 9
#define RED 10
void setup() {
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
}
void loop() {
  analogWrite(RED, 255);
  analogWrite(GREEN, 0);
  analogWrite(BLUE, 0);
  delay(1000);
  analogWrite(RED, 0);
  analogWrite(GREEN, 255);
  analogWrite(BLUE, 0);
  delay(1000);
  analogWrite(RED, 0);
  analogWrite(GREEN, 0);
  analogWrite(BLUE, 255);
  delay(1000);
}
```



Sekcja 7. Czujnik dźwięku

Przebieg

W tej lekcji dowiesz się, jak używać modułu czujnika dźwięku do sterowania diodą LED. Aby upewnić się, że mikrofon może normalnie wykrywać Twój głos, spróbuj zmienić jego czułość, obracając niebieskim precyzyjnym potencjometrem na module. Biorąc pod uwagę jego precyzję, potrzeba co najmniej 10 obrotów, aby uzyskać rzetelną odpowiedź.

Mikrofon

Przetworniki to urządzenia, które przekształcają energię z jednej postaci w drugą. Mikrofon to przetwornik, który przetwarza energię dźwięku na sygnały elektryczne. To działa odwrotnie do głośnika. Mikrofony są dostępne w różnych kształtach i rozmiarach. W zależności od zastosowania mikrofon może wykorzystywać różne technologie konwersji dźwięku na sygnały elektryczne. Tutaj omówimy elektretowy mikrofon pojemnościowy, który jest szeroko stosowany w telefonach komórkowych, laptopach itp. Jak sama nazwa wskazuje, elektretowy mikrofon pojemno-



Interfejs zapytania o pozyskiwanie lokalizacji

Pamiętaj, aby zezwolić urządzeniu na uzyskiwanie informacji o lokalizacji, w przeciwnym razie połączenie Bluetooth nie będzie możliwe!!!

Porady

Aplikacja uzyskuje informację o lokalizacji, a nie w celu monitorowania urządzenia ani uzyskania Twojej prywatności w telefonie. Ma na celu wygodniejsze i szybsze połączenie z Bluetooth.

Nasza aplikacja wykorzystuje najnowszą technologię „pop-up”, która jest zgodna z tradycyjną. W porównaniu z Bluetooth i aplikacją, nasza aplikacja nie wymaga ręcznego wybierania Bluetooth, wprowadzania hasła parowania, wystarczy zbliżyć telefon do samochodu, a Bluetooth wykona dynamiczne połączenie.

2. Włącz Bluetooth w telefonie

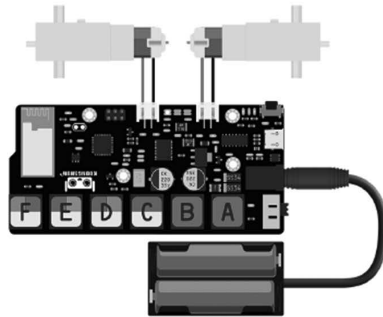
Ten krok jest bardzo prosty, ale nadal wymaga dopracowania: w większości telefonów komórkowych wystarczy kliknąć ikonę Bluetooth, aby włączyć Bluetooth, z wyjątkiem niektórych modeli Huawei. W Huawei oprócz włączenia Bluetooth, musisz także włączyć „Huawei Sharing”. Ponadto czasami telefony Huawei, ze względu na aktualizację systemu, nie mogą podłączyć się poprzez Bluetooth.

3. Wybór pojazdu

Po wejściu do aplikacji pojawia się interfejs wyboru samochodu. Obecnie opracowano łącznie 4 samochody, a samochód, który mamy, nazywa się „Tiny Smart Car”. Kliknij zdjęcie samochodu, aby przejść do następnego kroku. Kliknij ikonę w lewym górnym rogu, aby zmienić język. Obecnie aplikacja obsługuje 8 języków.

PIN:

zasilacz	Gniazdo zasilania płyty głównej
prawy silnik M2	Prawe gniazdo płyty głównej KS
lewy silnik M2	Lewe gniazdo płyty głównej KS



Pobieranie aplikacji:

IOS

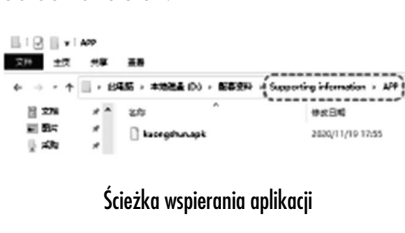
Otwórz APP Store i wyszukaj „Kuongsun”, aby znaleźć potrzebną aplikację. Kliknij „Pobierz”, aby dokończyć instalację aplikacji.



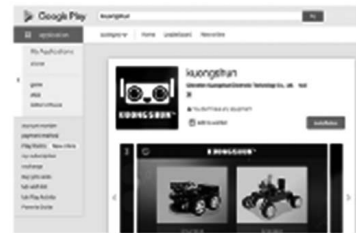
Schemat ideowy instalacji aplikacji systemu IOS

Android

Skopiuj "kuongsun.apk" w informacjach pomocniczych do telefonu komórkowego w celu instalacji aplikacji. Po prostu zainstaluj ją lub możesz otworzyć Google Play i wyszukać "kuongsun", aby znaleźć wymaganą aplikację. Kliknij "zainstaluj", aby pobrać na telefon.



Ścieżka wspierania aplikacji



Schemat instalacji aplikacji Google Play

Działanie aplikacji:

Połączenie Bluetooth

1. Uzyskanie pozwolenia

Po zainstalowaniu aplikacji, gdy jest otwierana po raz pierwszy, pojawi się pytanie czy zezwolić „kuongsun” uzyskać informacje o lokalizacji drugiego urządzenia?

ściowy jest równoległym kondensatorem płytowym i działa na zasadzie zmiennej pojemności. Składa się z dwóch płyt, jednej stałej (zwanej płytą tylną) i drugiej ruchomej (zwanej membraną) z niewielką szczeliną między nimi. Potencjał elektryczny ładuje płytkę. Kiedy dźwięk uderza w membranę, zaczyna się ona poruszać, zmieniając w ten sposób pojemność między płytami, co z kolei powoduje przepływ zmiennego prądu elektrycznego.

Mikrofony te są szeroko stosowane w obwodach elektronicznych do wykrywania drobnych dźwięków lub wibracji powietrza, które z kolei są przetwarzane na sygnały elektryczne do dalszego wykorzystania. Dwie nogi, jak pokazano na powyższym obrazku, służą do wykonania połączenia elektrycznego z obwodem.



Solidna, przewodząca metalowa obudowa otacza różne części mikrofonu. Górna powierzchnia jest pokryta porowatym materiałem za pomocą kleju. Działa jak filtr dla cząsteczek kurzu. Sygnały dźwiękowe/wibracje powietrza przechodzą przez porowaty materiał i spadają na membranę przez otwór pokazany na powyższym obrazku.

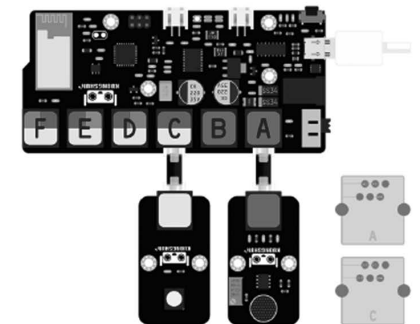
PIN:

PORT	PIN
A	A0
C	D3

KOD:

```
void setup() {
  Serial.begin(9600);
  pinMode(3, OUTPUT);
}

void loop() {
  Serial.println(analogRead(A0));
  if (analogRead(A0) >= 80) {
    digitalWrite(3, HIGH);
    delay(1000);
  }
  else {
    digitalWrite(3, LOW);
  }
}
```



Kod analogRead odczyta wartość analogową modułu i wyświetli ją bezpośrednio na monitorze szeregowym, podobnie, wartość tę można również zmienić zgodnie z metodą regulacji czułości wspomnianą powyżej.

Sekcja 8. Wentylator silnika i moduł RP

Przebieg

W tej lekcji nauczysz się używać potencjometru do sterowania prędkością wentylatora silnika.

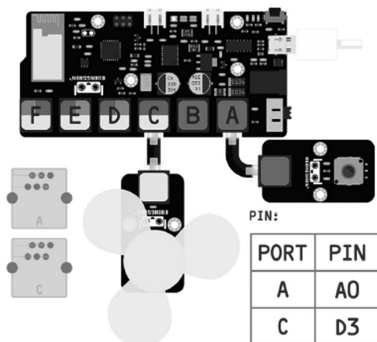
Potencjometr

Potencjometr to proste pokrętko, które zapewnia zmienną rezystancję. Obracając wałek potencjometru, zmieniamy wielkość rezystancji po obu stronach wycieraczki, która jest podłączona do środkowego bolca potencjometru. Zmienia to względ-

na „bliskość” tego styku na 5 woltów i masę, dając nam inne wejście analogowe. Kiedy wałek jest obrócony do końca w jednym kierunku, na kołek dochodzi 0 woltów i odczytujemy 0. Gdy wałek jest obrócony do końca w drugim kierunku, na kołek dochodzi 5 woltów i odczytujemy 1023. W międzyczasie funkcja analogRead() zwraca liczbę z przedziału od 0 do 1023, która jest proporcjonalna do wartości napięcia przykadanego do styku.

Silnik prądu stałego

Silnik prądu stałego (DC) jest najczęstszym typem silnika. Silniki prądu stałego zwykle mają tylko dwa przewody, jeden dodatni i jeden ujemny. Jeśli podłączysz te dwa przewody bezpośrednio do akumulatora, silnik zacznie się obracać. Jeśli zamienisz przewody, silnik będzie się obracał w przeciwnym kierunku.



Kod:

Najpierw musimy odczytać dane z potencjometru.

```
void setup() {
  Serial.begin(9600);
}

void loop(){
  val=analogRead(A0);
  Serial.println(map(val, 0, 1023, 0, 100));
}
```

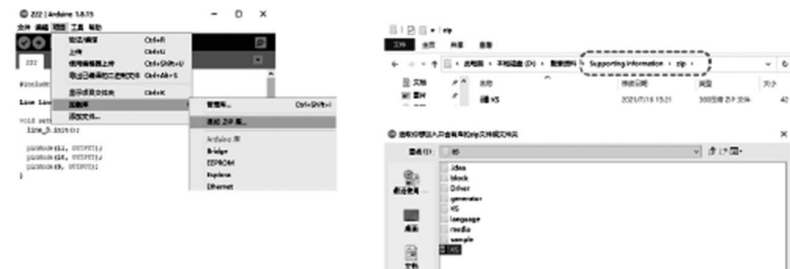
W programie analogRead(A0) służy do odczytu danych z potencjometru, a odczytywana wartość jest liczbą z przedziału 0-1023. Następnie musimy użyć funkcji map, aby przekonwertować ją proporcjonalnie na liczbę z zakresu 0-100. W ten sposób uzyskuje się wartość potencjometru w zakresie 0-100.

Wtedy możemy wykorzystać wartość potencjometru do sterowania wentylatorem. Wentylator musi używać portu PWM do sterowania prędkością wentylatora, a PWM ma wartość między 0-255, więc użyj map, aby zmienić wartość potencjometru na 0-255.

```
void setup() {
  Serial.begin(9600);
  pinMode(10, OUTPUT);
}

void loop(){
  Serial.println(map(analogRead, 0, 1023, 0, 100));
  analogWrite(10,(map(analogRead, 0, 1023, 0, 100) * 2,55));
}
```

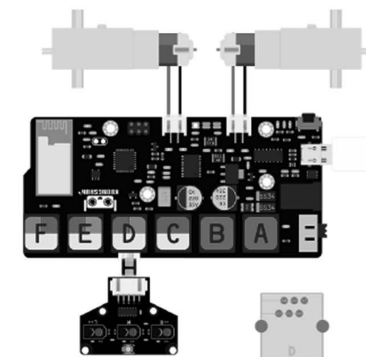
Pierwszym krokiem jest wybranie „Projekt” w górnym pasku menu, drugim krokiem jest wybranie „Załaduj bibliotekę”, trzecim krokiem kliknięcie „Dodaj bibliotekę .ZIP” i wreszcie znalezienie „KS.zip” w materiałach pomocniczych, aby ukończyć ładowanie.



```
KOD:
#include "KS.h"
Line line_d(0);
void yf_setMotor(int motorId, int motorDir,
int speed)
{
  int speedPin, directionPin;
  if (motorId == 1) // motor 1
  {
    speedPin = 5;
    directionPin = 6;
  } else if (motorId == 2) // motor 2
  {
    speedPin = 6;
    directionPin = 7;
  }
  else
  {
    return;
  }
  if (speed == 0)
  {
    digitalWrite(speedPin, 0);
  }
  else
  {
    if (motorDir == 0)
    {
      digitalWrite(directionPin, LOW);
      analogWrite(speedPin, speed);
    }
    else if (motorDir == 1)
    {
      digitalWrite(directionPin, HIGH);
      analogWrite(speedPin, speed);
    }
  }
}

void setup()
{
  line_d.init();
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
}

void loop()
{
  if (line_d.read(0))
  {
    yf_setMotor(1,0,100);
    yf_setMotor(2,0,100);
  }
  else
  {
    if (line_d.read(1))
    {
      yf_setMotor(1,1,150);
      yf_setMotor(2,1,200);
    }
    if (line_d.read(2))
    {
      yf_setMotor(1,0,200);
      yf_setMotor(2,1,150);
    }
  }
}
```



Sekcja 17. Moduł Bluetooth

Przebieg

W tej lekcji nauczysz się wykorzystywać moduł Bluetooth do sterowania samochodem.

Bluetooth

Bluetooth to technologia radiowa obsługująca komunikację na krótkie odległości między urządzeniami (zwykle w promieniu 10 m). Może bezprzewodowo wymieniać informacje między wieloma urządzeniami, w tym telefonami komórkowymi, palmtopami, bezprzewodowymi zestawami słuchawkowymi, notebookami i powiązаныmi urządzeniami peryferyjnymi. Wykorzystanie technologii „Bluetooth” może skutecznie uprościć komunikację pomiędzy mobilnymi urządzeniami końcowymi komunikacji. Nasz samochód zdalnie sterowany Bluetooth musi tylko podłączyć płytę główną, silnik i baterię. Oczywiście, jeśli chcesz, aby samochód był bardziej grywalny, możesz dodać funkcje brzęczyka, diody LED lub sygnalizacji świetlnej, jest to dozwolone. Wreszcie potrzebny jest również telefon komórkowy.

Sekcja 16. Moduł śledzenia trójkrotnego

Przebieg

W tej lekcji nauczysz się trójkierunkowego modułu śledzenia, aby sterować lewym i prawym silnikiem, tak by jechał prosto, w lewo i w prawo. Możemy zbudować samochód śledzący ze wszystkimi niezbędnymi komponentami.

Silnik

Silnik jest rodzajem urządzenia elektromagnetycznego, które może przetwarzać lub przesyłać energię elektryczną zgodnie z prawem indukcji elektromagnetycznej. Silnik w obwodzie jest oznaczony literą M, jego główną rolą jest wytwarzanie momentu napędowego, jako źródła zasilania urządzeń elektrycznych lub różnych maszyn.

Moduł śledzenia trójkrotnego

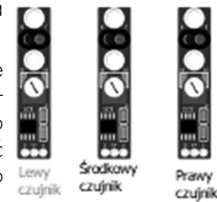
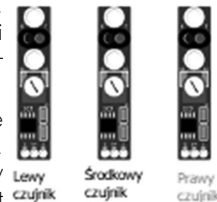
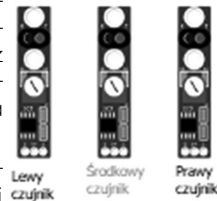
Jak sama nazwa wskazuje, moduł śledzenia będzie podążał ustaloną trasą. Oczywiście trasa tutaj polega na stworzeniu mapy obwodu z czarną taśmą elektryczną na podłodze, która kontrastuje czernią. Tutaj używamy taśmy elektrycznej do wyklejenia logo Arduino (∞), jednak możesz stworzyć własną ulubioną linię, ale ważne jest, aby pamiętać, że linia musi być. Powinna być niezbyt skomplikowana i niezbyt ostra.

Jak widać na zdjęciu, pojazd wyposażony jest w 3-kierunkowy czujnik śledzenia, czarna skrzynka to droga dla linii samochodowej. Gdy środkowy czujnik wykryje czarną linię, pojazd jedzie dalej po linii prostej, z dwoma samochodami po lewej i prawej stronie. Koła obracają się do przodu (upewnij się, że prędkość kół po obu stronach jest taka sama).

Gdy prawy czujnik wykryje czarną linię, oznacza to, że samochód zjechał z linii prostej w lewo lub skręcił w lewo. W tym momencie samochód musi być kontrolowany, aby przesunąć się lekko w prawo. Specyficzna operacja jest następująca: prędkość lewego koła do przodu jest wyższa niż prędkość prawego koła do tyłu.

Kiedy lewy czujnik wykryje czarną linię, oznacza to, że samochód zjechał z linii prostej w prawo lub skręcił w prawo. Musisz sterować samochodem tak, aby zjechał lekko w lewo. Specyficzna operacja jest następująca: prędkość prawego koła do przodu jest wyższa niż prędkość lewego koła do tyłu.

Ponieważ kod korzysta z naszej własnej biblioteki, musimy załadować bibliotekę w informacjach pomocniczych. Poniżej przedstawiono metodę ładowania biblioteki.



PIN:

PORT	PIN	M2(R)	M1(L)
D	D8	Ⓟ Lewy interfejs	Ⓟ Prawy interfejs

Wgraj program do Arduino i obracaj potencjometrem, aby zmienić prędkość silnika. Jeśli silnik nie może się normalnie uruchomić, należy lekko obrócić topatki wentylatora ręcznie.

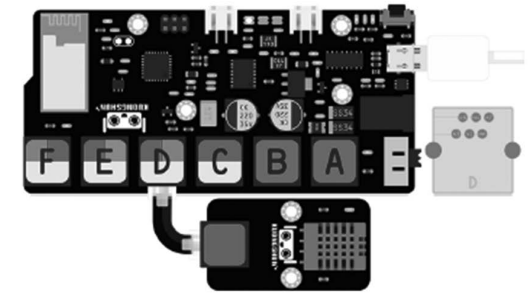
Sekcja 9. Czujnik DHT11

Przebieg

W tym samouczku nauczymy się, jak korzystać z czujnika temperatury i wilgotności DHT11. Jest wystarczająco dokładny dla większości projektów, które muszą śledzić odczyt wilgotności i temperatury.

PIN:

PORT	PIN
D	D8



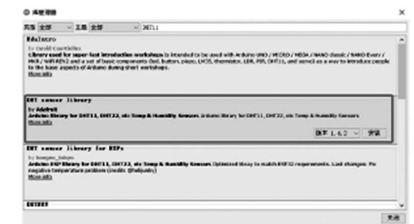
Kod

Aby odczytać program temperatury i wilgotności DHT11, musisz dodać bibliotekę modułów DHT11 w arduino.

Otwórz arduino, aby znaleźć bibliotekę zarządzania:



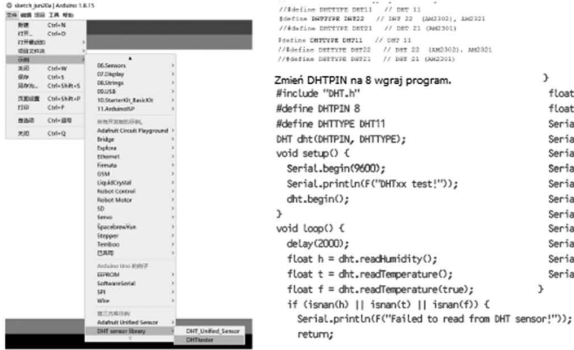
Wyszukaj DHT11, zainstaluj tę bibliotekę Adafruit.



Kliknij zainstaluj wszystko.



Następnie znajdź program w przykładzie.



Odhacz DHT22 i zmień na DHT11.

Sekcja 10. Moduł fotorezystora

Przebieg

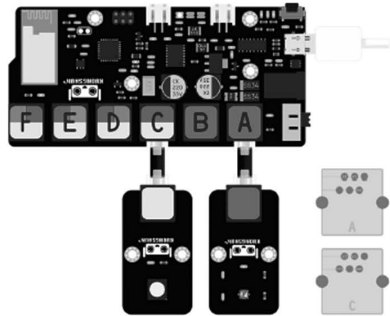
W tej lekcji dowiesz się, jak używać wejścia analogowego do pomiaru natężenia światła w celu sterowania jasnością diod LED.

PIN:

PORT	PIN
C	D3
A	A0

KOD:

```
void setup() {
  Serial.begin(9600);
  pinMode(3, OUTPUT);
}
void loop() {
  Serial.println(analogRead(A0));
  if (analogRead(A0) > 650) {
    digitalWrite(3,HIGH);
  } else {
    digitalWrite(3,LOW);
  }
}
```



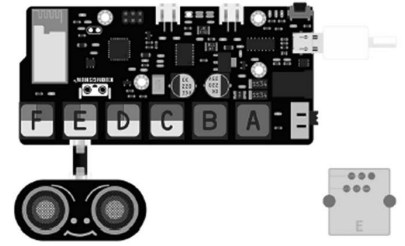
W programie, gdy ustawimy wartość światła na większą niż 650, dioda będzie świecić. 650 tutaj jest tylko wartością odniesienia. Możesz ustawić ją zgodnie z bieżącym środowiskiem. Im silniejsze światło, tym mniejsza wartość.

PIN:

PORT	PIN
E	D10

KOD:

```
const int TrigPin = 10;
const int EchoPin = 9;
float cm;
void setup() {
  Serial.begin(9600);
  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
}
void loop() {
  digitalWrite(TrigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(TrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(TrigPin, LOW);
  cm = pulseIn(EchoPin, HIGH) /
  58.0;
  cm = (int(cm * 100.0)) / 100.0;
  Serial.print(cm);
  Serial.print("\n");
  Serial.println();
  delay(1000);
}
```



Sekcja 15. Moduł Micro Servo 9g

Przebieg

W tej lekcji dowiesz się, jak używać modułu ultradźwiękowego do kontrolowania kąta obrotu przekładni kierowniczej.

Serwo

Przekładnia kierownicza to serwonapęd pozycyjny (kątowy) do systemów sterowania, w których kąty stale się zmieniają i mogą być utrzymywane. Nadaje się głównie do systemów sterowania, które muszą stale zmieniać kąt, takich jak ramiona i nogi robotów humanoidalnych, sterowanie kierunkiem modeli samochodów i modeli samolotów. Sygnał sterujący przekładni kierowniczej jest w rzeczywistości sygnałem modulacji szerokości impulsu (sygnał PWM).

W tej lekcji moduł ultradźwiękowy jest używany do wykrywania różnych odległości w celu kontrolowania kąta obrotu.

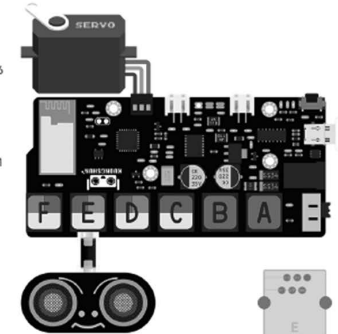
KOD:

```
#include <Servo.h>
Servo servo_11;
float checkdistance_10_9()
{
  digitalWrite(10, LOW);
  delayMicroseconds(2);
  digitalWrite(10, HIGH);
  delayMicroseconds(10);
  digitalWrite(10, LOW);
  float distance = pulseIn(9, HIGH)
  / 58.00;
  delay(10);
  return distance;
}
void setup() {
  pinMode(10, OUTPUT);
  pinMode(9, INPUT);
  servo_11.attach(11);
}
void loop() {
  if (checkdistance_10_9() > 10 &&
```

PIN:

PORT	PIN	Linia serw
E	D10	Błęszona Pomarańczowa Czerwona

Wewnętrzne nagłówki pinów są wstawiane w kolejności



Prześlij kod do Arduino i poruszaj joystickiem, możesz zauważyć, że kolor i jasność RGB zmieniają się wraz z kierunkiem joysticka.

Sekcja 14. Moduł ultradźwiękowy

Przebieg

Czujnik ultradźwiękowy doskonale nadaje się do wszelkiego rodzaju projektów wymagających pomiaru odległości.

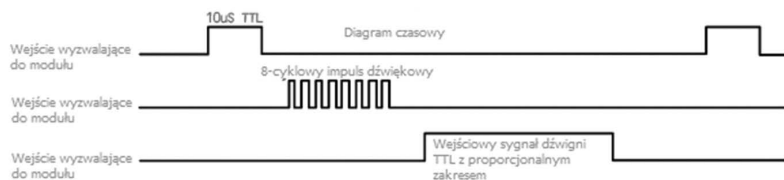
Czujnik ultradźwiękowy

Moduł czujnika ultradźwiękowego HC-SR04 zapewnia bezkontaktową funkcję pomiaru 2 cm - 400 cm, dokładność pomiaru odległości może osiągnąć 3 mm. Moduły zawierają nadajniki ultradźwiękowe, odbiornik i obwód sterujący.

Podstawowa zasada pracy:

1. Używając wyzwalacza IO dla sygnału wysokiego poziomu co najmniej 10µs.
2. Moduł automatycznie wysyła osiem 40 kHz i wykrywa, czy jest zwrotny sygnał impulsowy.
3. JEŚLI sygnał powraca, przez wysoki poziom, czas trwania IO o wysokim poziomie wyjściowym jest czasem od wystąpienia ultradźwiękowego obracania skrętów.

Odległość testowa = (czas wysokiego poziomu x prędkość dźwięku (340 m/s)/2). Wykres czasowy pokazano poniżej. Wystarczy podać krótki impuls 10µs na wejście wyzwalające, aby rozpocząć określanie zakresu, a następnie moduł wyśle 8-cykłowy impuls ultradźwięków o częstotliwości 40 kHz i podniesienie jego echa. Echo jest obiektem odległości, który jest proporcjonalny do szerokości impulsu i zasięgu. Zasięg można obliczyć na podstawie przedziału czasu między wystąpieniem sygnału wyzwalającego, a odebraniem sygnału echa. Wzór: $1\mu s / 58 = \text{centymetr}$ lub $1\mu s / 148 = \text{cal}$, lub: $\text{zakres} = \text{czas wysokiego poziomu} * \text{prędkość} (340\text{m/s}) / 2$. Sugerujemy użycie cyklu pomiarowego powyżej 60 ms, aby zapobiec wysłaniu sygnału wyzwalającego do echosygnatu.



Sekcja 11. Moduł pirolizy człowieka

Przebieg

W tej lekcji dowiesz się, jak używać czujnika ruchu PIR z UNO. UNO jest sercem tego projektu. „Nastłuchuje” czujnika PIR, a po wykryciu ruchu instruuje diodę LED, aby się zaświeciła lub zgasła.

Czujnik PIR

Czujniki PIR są bardziej skomplikowane niż wiele innych czujników opisanych w tym samouczku (takich jak fotokomórki, FSR i przełączniki pochylenia), ponieważ istnieje wiele zmiennych, które wpływają na wejście i wyjście czujnika. Sam czujnik PIR ma dwa gniazda. Każde gniazdo jest wykonane ze specjalnego materiału, który jest wrażliwy na IR. Zastosowany tutaj obiektyw tak naprawdę nie robi wiele, więc widzimy, że dwa gniazda „widzą” z pewnej odległości (w zasadzie czułość czujnika widzi). Gdy czujnik jest bezczynny, oba gniazda wykrywają tę samą ilość promieniowania podczerwonego, czyli ilość promieniowania otoczenia emitowanego z pomieszczenia, ścian lub na zewnątrz. Kiedy przechodzi ciepłe ciało, takie jak człowiek lub zwierzę, najpierw przechwytuje jedną połowę czujnika PIR, co powoduje dodatnią zmianę różnicy między dwiema połówkami. Kiedy ciepłe ciało opuszcza obszar wykrywania, następuje odwrotna sytuacja, w wyniku której czujnik generuje ujemną zmianę różnicy. Te impulsy zmian są tym, co jest wykrywane.

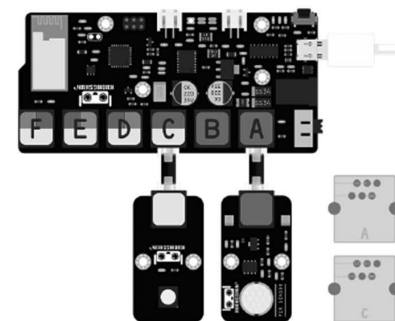
PIN:

PORT	PIN
A	A0
C	D3

KOD:

```
void setup(){
  Serial.begin(9600);
  pinMode(3, OUTPUT);
}

void loop(){
  Serial.println(analogRead(A0));
  if (analogRead(A0) > 500) {
    digitalWrite(3,HIGH);
  } else {
    digitalWrite(3,LOW);
  }
}
```



Sekcja 12. Moduł dotykowy

Przebieg

Zasada działania modułu dotykowego i przycisku jest taka sama. Włącznik diody LED może być sterowany za pomocą modułu dotykowego.

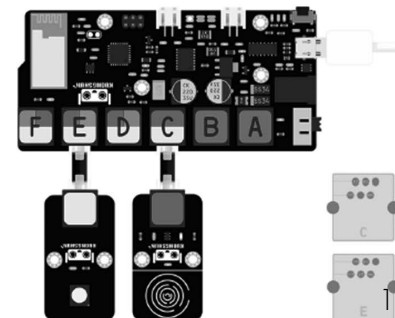
PIN:

PORT	PIN
E	D10
C	D3

KOD:

```
void setup() {
  pinMode(3, INPUT);
  pinMode(10, OUTPUT);
}

void loop() {
  if (digitalRead(3)) {
    digitalWrite(10, HIGH);
  } else {
    digitalWrite(10, LOW);
  }
}
```



Sekcja 13. Moduł joysticka

Przebieg

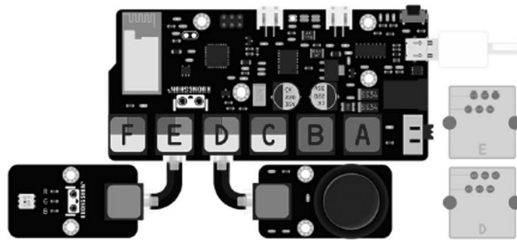
W tej lekcji nauczysz się używać joysticka do sterowania kolorem i jasnością RGB.

Pilot na podczerwień

Myszę, że każdy grał w gry. W końcu granie w gry jest o wiele przyjemniejsze niż nauka, ale nie wiem, czy ktoś używa gamepada do grania w gry. Joystick bardziej przypomina gamepad, kontrolując nasze światła RGB pod różnymi kątami i w różnych kierunkach. W porządku, jeśli nie grałeś, zagrajmy razem w grę, ale zamiast grać w grę, używamy teledetekcji do sterowania światłami RGB. Celem jest umożliwienie każdemu nauczyciela się odczytywania wartości z joysticka i sterowania nim za pomocą joysticka.

PIN:

PORT	PIN
E	D9, D10, D11
D	A2
	A3



Porady

Joystick domyślnie ustawiony jest do prawej ręki do sterowania nim! Jeśli niektórzy uczniowie są przyzwyczajeni do sterowania leworęcznego, należy zamienić wartości osi X i Y.

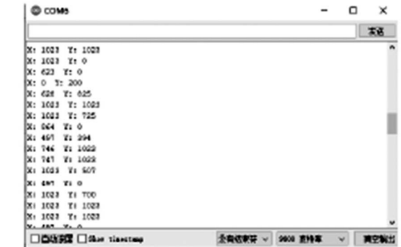
Przed rozpoczęciem gry musimy wykonać pewne prace przygotowawcze, najpierw odczytać wartość pozycji joysticka X, Y.

Odczytaj kod wartości X, Y:

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Serial.print("X: ");
  Serial.print(analogRead(A2));
  Serial.print(" ");
  Serial.print("Y: ");
  Serial.println(analogRead(A3));
  delay(100);
}
```

Otwórz monitor szeregowy i potrząśnij joystickiem, zobaczysz, że wartości X i Y na monitorze szeregowym stale się zmieniają.

Monitor szeregowy do wyświetlania danych joysticka



Po odczytaniu osi X i Y możemy oceniać i kontrolować zgodnie z odczytaną wartością, należy jednak zaznaczyć, że wartość graniczna oceny powinna być nieco wyższa i nieco niższa od odczytanej wartości.

Porady

Dane odczytane tutaj nie są takie same jak dane, o których wspominaliśmy powyżej. Powyższy schemat wykorzystuje port szeregowy do odczytania wartości początkowej, ponieważ nie znamy wartości X i Y. Odczytana tutaj wartość ma pozwolić uzyskać wartości sterować światłem RGB.

Odczytaj kod danych joysticka:

```
volatile float X;
volatile float Y;
void setup()
{
  X = 0;
  Y = 0;
}
void loop()
{
  X = analogRead(A2);
  Y = analogRead(A3);
}
```

Kod światła RGB sterowania osią X:

```
volatile float X;
volatile float Y;
void setup()
{
  X = 0;
  Y = 0;
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
}
void loop()
{
  X = analogRead(A2);
  Y = analogRead(A3);
  if (X >= 505)
  {
    analogWrite(10,X);analog-
```

Porady

Należy zauważyć, że: odczytujemy wartość X przez port szeregowy 497, ale w celu zmniejszenia błędów, zwiększyliśmy wartość graniczną do 505 i 492 i możemy ją zmodyfikować zgodnie z rzeczywistą sytuacją.

Kod światła RGB sterowania osią Y:

```
volatile float X;
volatile float Y;
void setup()
{
  X = 0;
  Y = 0;
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
}
void loop()
{
  X = analogRead(A2);
  Y = analogRead(A3);
  if (Y >= 505)
  {
    analogWrite(10,0);analog-
```

Kod kolekcji X, Y:

```
volatile float X;
volatile float Y;
void setup()
{
  X = 0;
  Y = 0;
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
}
void loop()
{
  X = analogRead(A2);
  Y = analogRead(A3);
  if (X >= 505)
  {
    analogWrite(10,X);analog-
```