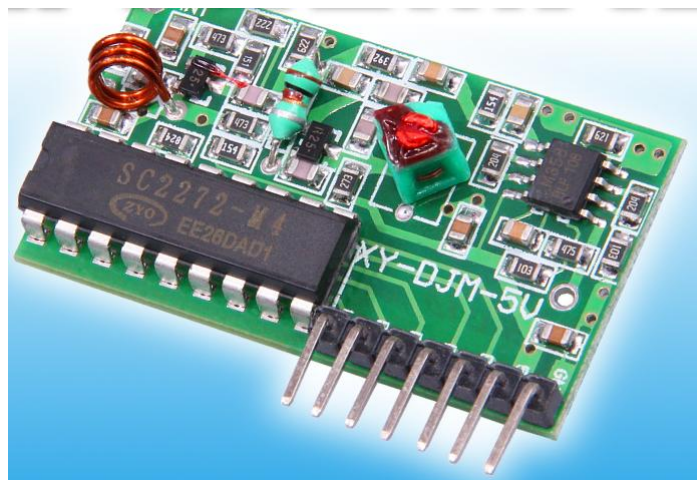


Pilot RF 4-kanalowy + odbiornik XY-DJM-5V
umożliwia zdalne sterowanie do czterech urządzeń.



Właściwości:

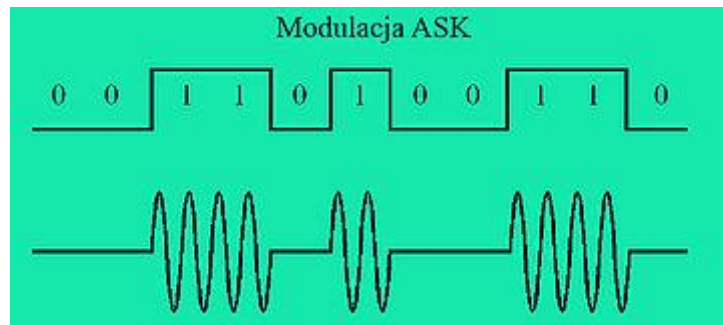
Nadajnik pilot MX804

- zasilanie pilota bateria L1028 23A 12V
- Napięcie zasilające 3V do 12 V
- Pobierany prąd $\text{Max} \leq 40\text{mA}$ (12V), $\text{Min} \leq 9\text{mA}$ (3V)
- Częstotliwość 315MHz~433.92MHz (433,050-434,790 MHz)
- Błąd częstotliwości $\pm 150\text{kHz}(\text{max})$
- Modulacja ASK/OOK
- Szybkość transmisji $\leq 10\text{Kbps}$
- Moc transmisji 25mW (315MHz@12V)
- możliwość zmiany kodu poprzez zworki przy układzie PT2262

Odbiornik XY-DJM

- chip PT2272 (SC2272)
- zasilanie 5V
- możliwość zmiany kodu poprzez zworki przy układzie PT2272
- wyprowadzenia dla kanałów A, B, C, D, ABCD
- przytrzymanie przycisku na pilocie powoduje pojawienie się stanu H na wyjściu odbiornika do czasu zwolnienia przycisku

Modulacja ASK jest odpowiednikiem analogowej modulacji [AM](#). Fala nośna zwiększa lub zmniejsza swoją amplitudę w zależności od zmieniającego się ciągu bitów. W najprostszym przypadku logiczne 0 jest reprezentowane jako brak nośnej (amplituda równa 0) natomiast logiczne 1 jest sygnałem harmonicznym o określonej amplitudzie. W ASK faza oraz częstotliwość nośnej nie podlega żadnej zmianie. Poglądowo przedstawia to poniższy rysunek:



Taki sposób modulacji określa się często pojęciem OOK (*on-off keying*) i był stosowany w radiotelegrafii wykorzystującej alfabet Morse'a.

Przykład programu

```
void setup()
{
  pinMode(13,OUTPUT);
}
void loop()
{
  if (digitalRead(7) == HIGH)
    digitalWrite(13, HIGH);
  else
    digitalWrite(13, LOW);
}
```

Zadanie:

Napisać program, który pozwala na sterowanie świeceniem diody LED. Przycisk A powinien umożliwić zapalenie/zgaszenie diody zielonej, przycisk B – diody żółtej, przycisk C – diody czerwonej. Wciśnięcie przycisku D powinno powodować zapalenie/zgaszenie wszystkich trzech diod.

Zadanie:

Napisać program, który pozwala na sterowanie kierunkiem obrotów silnika krokowego za pomocą pilota RF. Sprawdzić możliwość występowania zakłóceń od sąsiednich nadajników

Pilot IR zdalnego sterowania standardu NEC + odbiornik z płytka PCB. Idealne rozwiązanie do rozbudowy swoich projektów o zdalne sterowanie. Obsługiwany między innymi przez bibliotekę ARDUINO.

Pilot IR:

Liczba klawiszy - 21

- Zasilanie - bateria CR2025 3V
- Standard kodowania - NEC
- Częstotliwość nośnej - 38KHz
- Zasięg transmisji: 8m
- Efektywny kąt: 60 stopni
- Prąd spoczynkowy 3-5uA, dynamiczny 3-5mA

Odbiornik IR:

- Nazwa / Model - VS1838 B
- Częstotliwość nośnej - 38KHz
- Zasilanie - min 2.7V, max 5.5V
- Temp pracy -20 ~ +80
- Płytkę PCB z diodą LED sygnalizującą

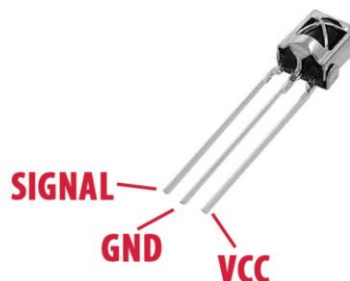
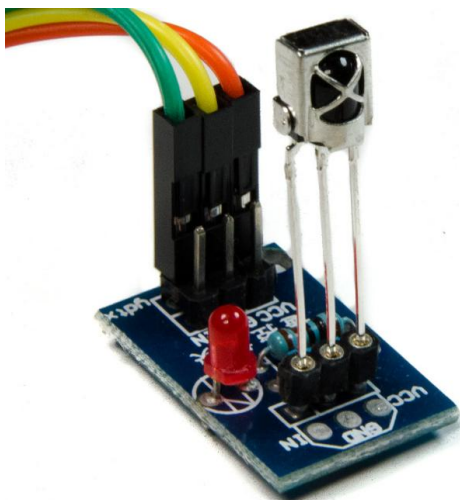
W zestawie:

Pilot IR (21 klawiszy)

- bateria CR2025 3V
- Odbiornik IR VS1838 B
- Płytkę PCB ze złączem na odbiornik
- Przewód 3 żyłowy ze złączem goldpin żeńskim

Parametry techniczne:

- Ø Zasilanie CR2025
- Ø Modulacja **HX1838**
- Ø Zasięg **do 8m**
- Ø **21 przycisków!**



Przykładowy program

```
/*
 * IRremote: IRrecvDemo - demonstrates receiving
 * IR codes with IRrecv
 * An IR detector/demodulator must be connected
 * to the input RECV_PIN.
 * Version 0.1 July, 2009
 * Copyright 2009 Ken Shirriff
 * http://arcfn.com
 */

#include <IRremote.h>

int RECV_PIN = 11;      //SEE NOTE in Sheepdogguides text
//Must be set to the Arduino pin your sensor is on.
//http://sheepdogguides.com/arduino/ar3ne1ir.htm

IRrecv irrecv(RECV_PIN);

decode_results results;

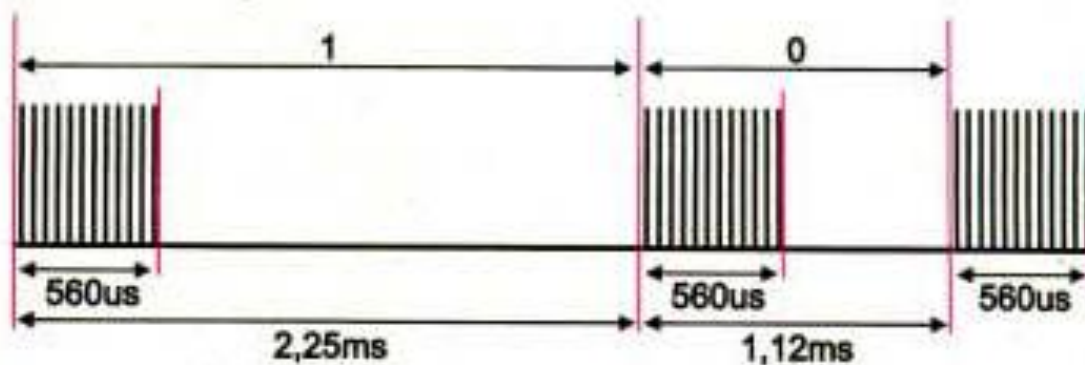
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    Serial.println(results.decode_type, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

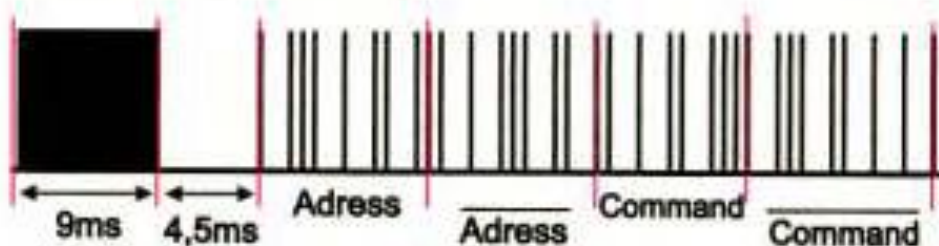
Zadanie:

Napisać program, który pozwala na sterowanie diodami LED podobnie jak w przypadku pilota radiowego.

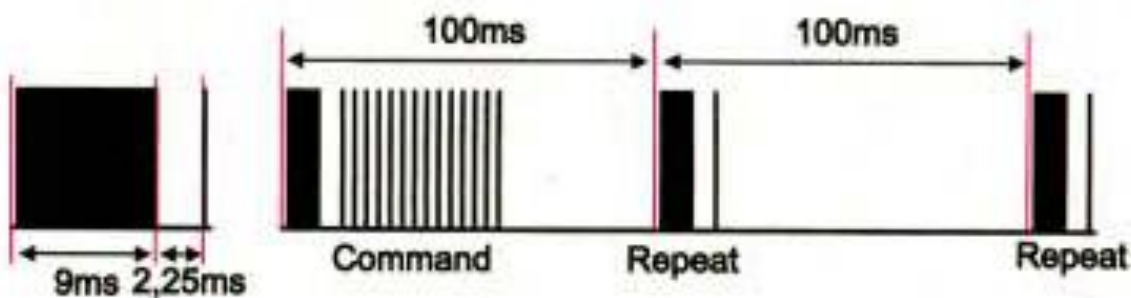
Modulacja



Protokół



Kod powtarzania



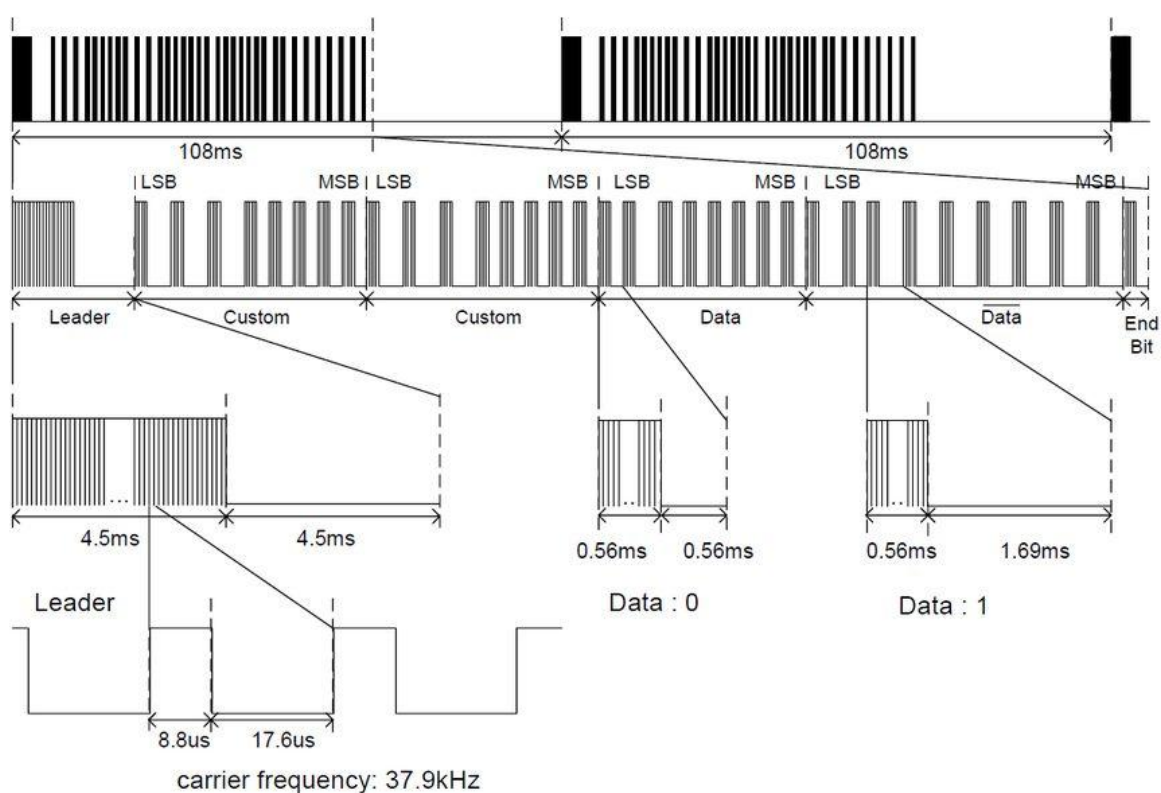
Standard NEC zakłada następujący protokół:

1. Stan wysoki – 9ms
2. Stan niski – 4.5ms
3. 8 bitów adresu
4. 8 zanegowanych bitów adresu
5. 8 bitów komendy
6. 8 zanegowanych bitów komendy

“1” wysyłana jest jako 560us stanu wysokiego i 1690us stanu niskiego, co daje łączną długość bitu 2.25ms.

“0” wysyłane jest jako 560us stanu wysokiego i 560us stanu niskiego, co daje łączną długość bitu 1.12ms.

W przypadku przytrzymania przycisku, polecenie nie jest powtarzane, zamiast tego wysyłany jest 9 milisekundowy stan wysoki, po którym następuję 2.25ms sekund przerwy i ponownie 560us stanu wysokiego. Sekwencja ta wysyłana jest co 110ms, tak długo, jak wciśnięty pozostaje przycisk.



Więcej o protokole NEC (i innych), można przeczytać pod adresem:

<http://www.sbprojects.com/knowledge/ir/nec.php>

<https://en.wikipedia.org/wiki/RC-5>

RC-5 PHILIPS PROTOCOL

