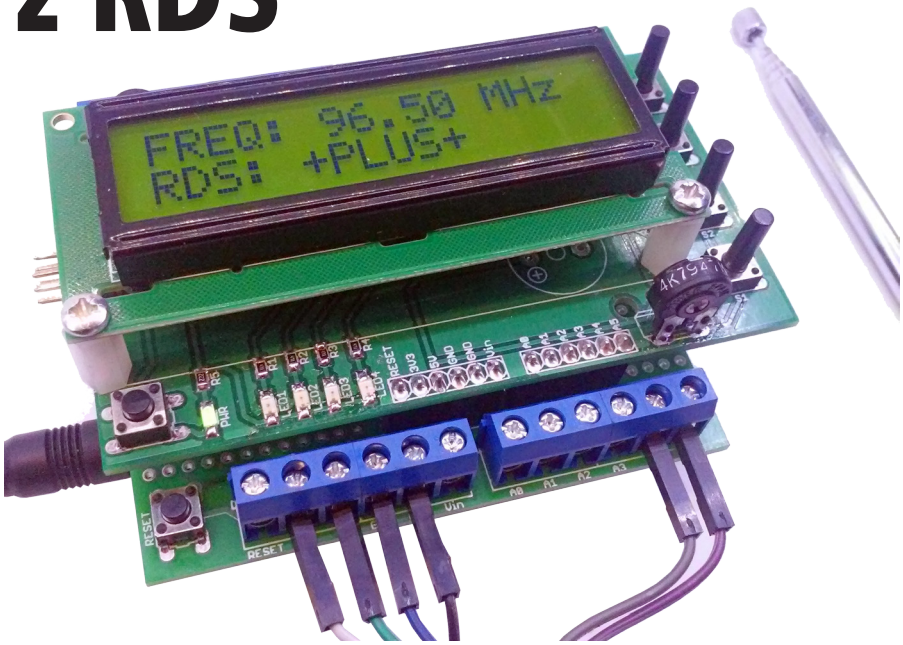
Radio FM z RDS

Jak samodzielnie i łatwo, bez nawijania cewek i strojenia obwodów, zbudować odbiornik radiowy? Odpowiedź jest jedna – należy zastosować odpowiedni moduł!

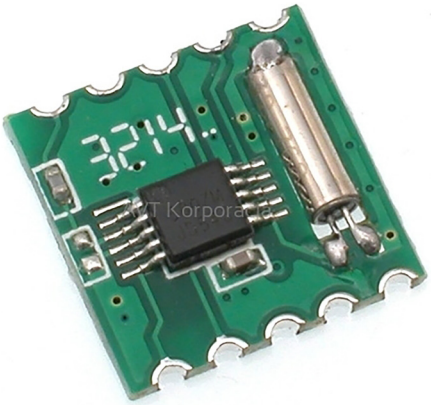


Prezentowane radio powstało na bazie modułu z układem RDA5807, jego najważniejsze cechy to:

- Jednoukładowy, kompletny tuner radiowy.
- Odbiór informacji RDS.
- Stereofoniczny.
- Funkcje redukcji szumów, podbicia basów, regulacji głośności.
- Interfejs I²C, zasilanie 3,3 V.

Podstawową aplikację układu tunera pokazano na **rysunku 1**. Do prawidłowej pracy wymaga on dosłownie kilku elementów zewnętrznych. Moduł z układem widoczny jest na **fotografii 2**. Kłopotliwy może okazać się niestandardowy raster wyprowadzeń, ale można poradzić sobie dolutowując listwy golpinów z lekko zgiętymi szpilkami, jak pokazano na **fotografii 3**. Na **rysunku 4** pokazano rozmieszczenie wyprowadzeń modułu.

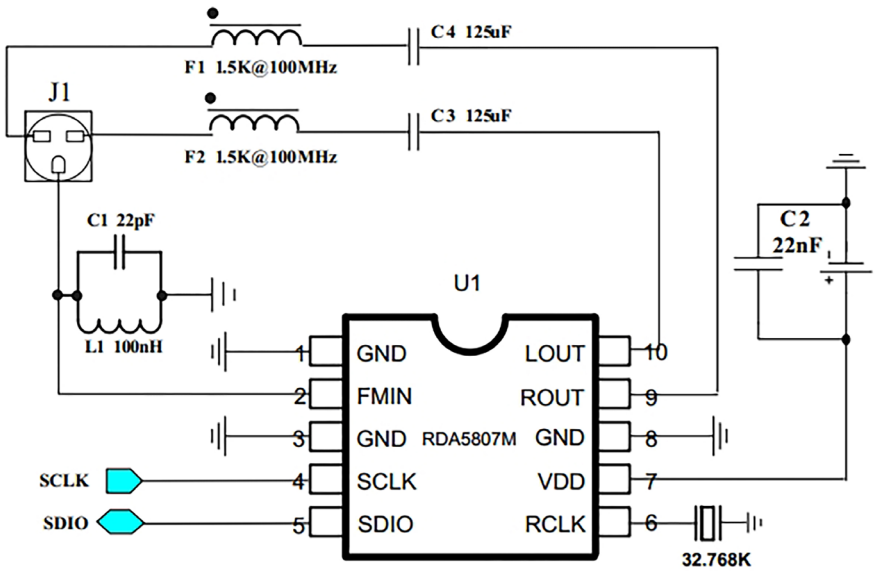
Układ RDA5807 pozwala na dołączenie słuchawek do wyjścia audio, wtedy kabel słuchawkowy może pełnić rolę anteny. Aby uzyskać większą moc sygnału audio w projekcie zastosowano dodatkowy wzmacniacz mocy – moduł AVT1498 (**fotografia 5**).



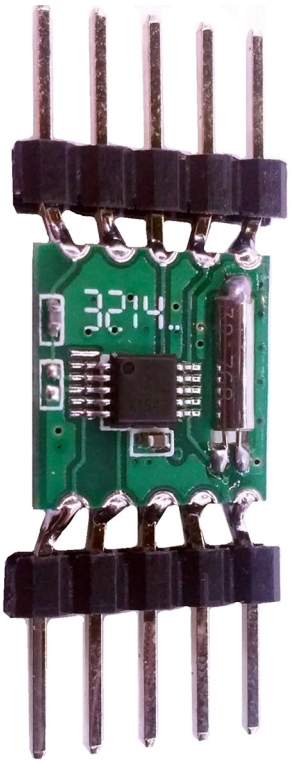
Fotografia 2. Widok modułu

Listing 1. Lista dostępnych komend i funkcji

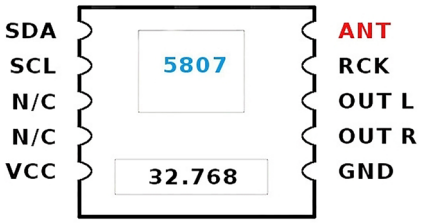
```
if (cmd == '?') {
    Serial.println();
    Serial.println("? Help");
    Serial.println("+ increase volume");
    Serial.println("- decrease volume");
    Serial.println("> next preset");
    Serial.println("< previous preset");
    Serial.println(". scan up   : scan up to next sender");
    Serial.println(", scan down ; scan down to next sender");
    Serial.println("fnnnnn: direct frequency input");
    Serial.println("i station status");
    Serial.println("s mono/stereo mode");
    Serial.println("b bass boost");
    Serial.println("u mute/unmute");
}
```



Rysunek 1. Podstawowa aplikacja układu RDA5807



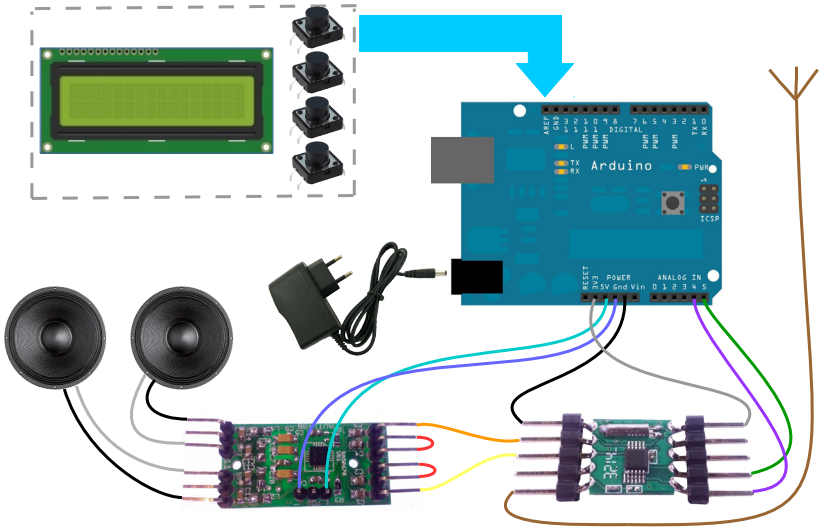
Fotografia 3. Sposób montażu goldpinów



Rysunek 4. Rozmieszczenie wyprowadzeń modułu

Funkcję anteny pełni typowa antena teleskopowa.

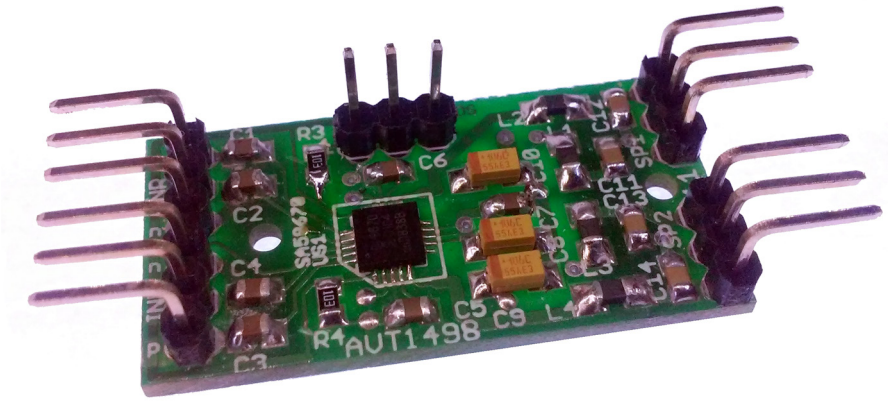
Sterowanie modułem zrealizowano na bazie płytki Arduino Uno z dołączoną płytką interfejsu użytkownika AVT1615. Dla wygody zestaw został uzupełniony o płytkę ze złączami – AVT1633. Elementy zestawu pokazano na **fotografii 6**. Sposób połączenia wszystkich komponentów zaprezentowano



Rysunek 7. Sposób połączenia komponentów urządzenia

na rysunku 7 i dla ułatwienia, rzeczywisty wygląd na **fotografii 8**.

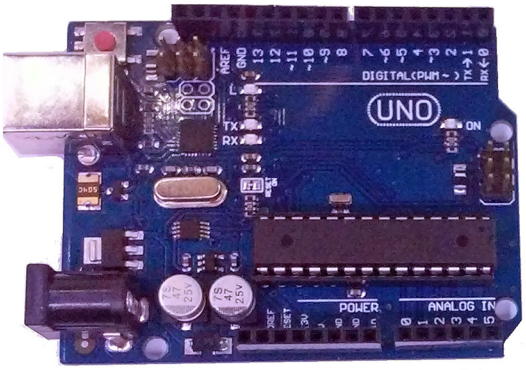
Program sterujący pracą urządzenia powstał w środowisku Arduino IDE i bazuje na jednym z przykładów dołączonych do dokumentacji modułu. Chodzi o projekt

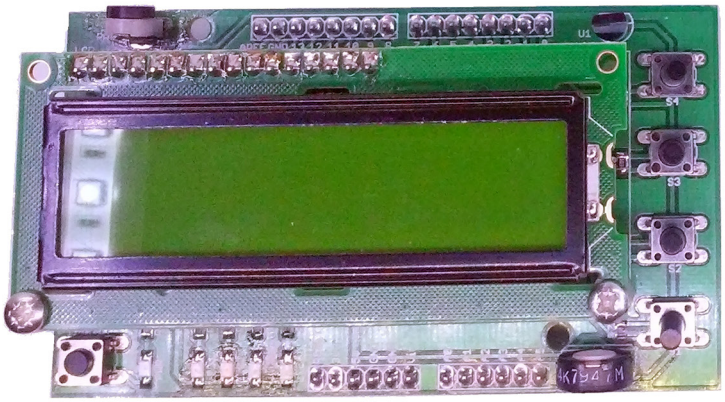


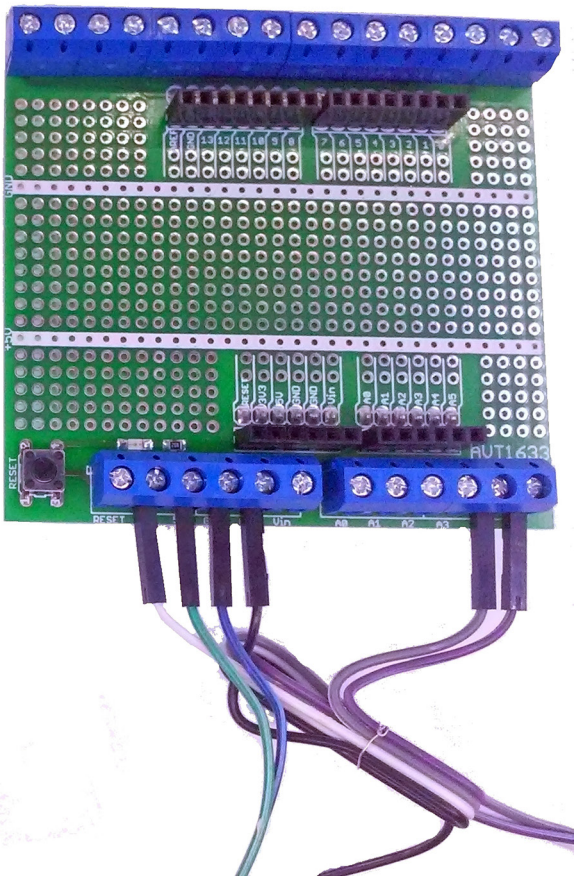
Fotografia 5. Moduł wzmacniacza audio AVT1498

Listing 2. Przypisanie funkcji wyprowadzeniom płytki Arduino

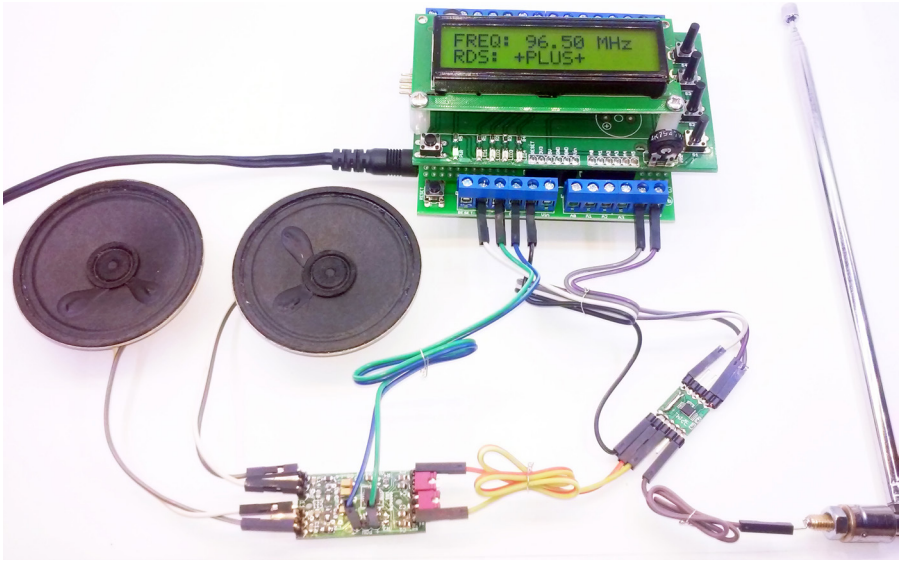
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
int button1_pin = 3;
int button2_pin = 2;
int button3_pin = 1;
int button4_pin = 0;
int button;  //zmienna pomocnicza
```



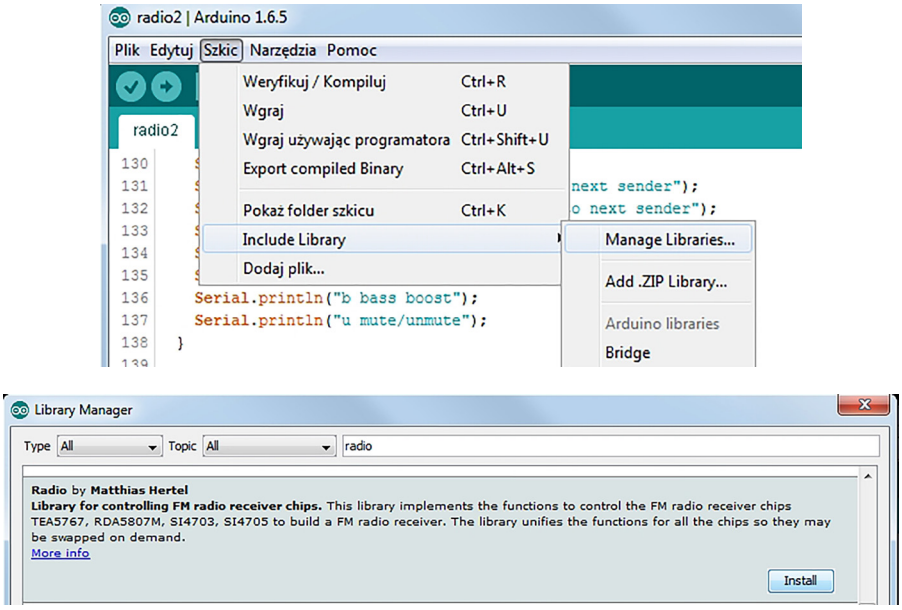




Fotografia 6. Moduły rozszerzeń dla płytki Arduino



Fotografia 8. Widok połączonego zestawu



Rysunek 9. Sposób dołączenia biblioteki Radio

SerialRadio, który pozwala na uruchamianie wszystkich funkcji modułu poprzez komendy wysyłane z terminala. Lista dostępnych komend i funkcji jest wyświetlana po wysłaniu znaku zapytania do układu, jak na **listingu 1**. Jednak zanim projekt zostanie skompilowany, należy dołączyć dodatkową bibliotekę „Radio". W tym celu należy postępować zgodnie ze wskazówkami z **rysunku 9**.

Do projektu należy dodać obsługę wyświetlacza oraz przycisków dostępnych w płytce AVT1615. Przed wywołaniem funkcji *setup()* należy dodać fragment z **listingu 2**, natomiast wewnątrz tej funkcji fragment z listingu 3. W ten sposób przypisujemy wyprowadzeniom płytki Arduino odpowiednie funkcje, inicjujemy wyświetlacz oraz wyłączmy port szeregowy, który zakłócałby działanie przycisków. Kolejne zmiany mają za zadanie przeniesienie na wyświetlacz informacji wysyłanych do portu szeregowego. Należy zmodyfikować funkcje *DisplayFrequency()* oraz *DisplayServiceName()* jak na **listingu 4**. Ostatnia zmiana to umieszczenie na końcu programu funkcji sprawdzającej stan przycisków *ButtonCheck()* oraz wywoływanie tej funkcji wewnątrz pętli głównej *loop()* jak na **listingu 5**. Dzięki modyfikacjom po przyciśnięciu któregoś przycisku program wykona funkcje jak gdyby otrzymał komendę z terminala *runSerialCommand()*. W tym przypadku przyciski powodują zwiększenie/zmniejszenie głośności oraz przełączenie na kolejną/poprzednią stacje z listy. Warto jeszcze uzupełnić listę stacji wpisując wartości częstotliwości ulubionych stacji (**listing 6**).

Pełne źródła programów znajdują się w materiałach dodatkowych dołączonych do projektu.

KS

Listing 3. Inicjalizacja elementów interfejsu
```
// open the Serial port
//Serial.begin(57600);
//Serial.print(„Radio...");

lcd.begin(16,2);
lcd.print(„Hello");
delay(500);

pinMode(button1_pin, INPUT_PULLUP);
pinMode(button2_pin, INPUT_PULLUP);
pinMode(button3_pin, INPUT_PULLUP);
pinMode(button4_pin, INPUT_PULLUP);
```

Listing 4. Modyfikacja funkcji przesyłających informacje
```
/// Update the Frequency on the LCD
display.
void DisplayFrequency(RADIO_FREQ f)
{
    char s[12];
    radio.formatFrequency(s,
sizeof(s));
    //Serial.print(„FREQ:"); Serial.
println(s);
    lcd.setCursor(0, 0);
    lcd.print(„FREQ:");
    lcd.print(s);
} // DisplayFrequency()

/// Update the ServiceName text
on the LCD display.
void DisplayServiceName(char *name)
{
//Serial.print(„RDS:");
//Serial.println(name);
    lcd.setCursor(0, 1);
    lcd.print(„RDS:");
    lcd.print(name);
} // DisplayServiceName()
```

Listing 5. Przypisanie funkcji przyciskom
```
//sprawdzenie przycisków
button = ButtonCheck();
if (button > 0) {
    //refresh = 1;
    if (button == 1) {
        runSerialCommand('>', 0);
    }
    if (button == 2) {
        runSerialCommand('<', 0);
    }
    if (button == 3) {
        runSerialCommand('+', 0);
    }
    if (button == 4) {
        runSerialCommand('-', 0);
    }
    button = 0;
}
```

Listing 6. Przykładowa lista częstotliwości stacji radiowych
```
RADIO_FREQ preset[] = {
    9100,
    9650,
    10100,
    10300,
    10560,
    10750
};
```