

QAPRSTracker

Łukasz Nidecki SQ5RWU

Spis treści

QAPRSTracker – opis urządzenia.....	3
Zasada działania.....	5
Odbiór.....	5
Nadawanie.....	5
Zasilanie.....	6
GPS.....	6
Wyświetlacz.....	6
Karta SD.....	6
Klawiatura.....	7
Konfiguracja.....	8
Oprogramowanie.....	9

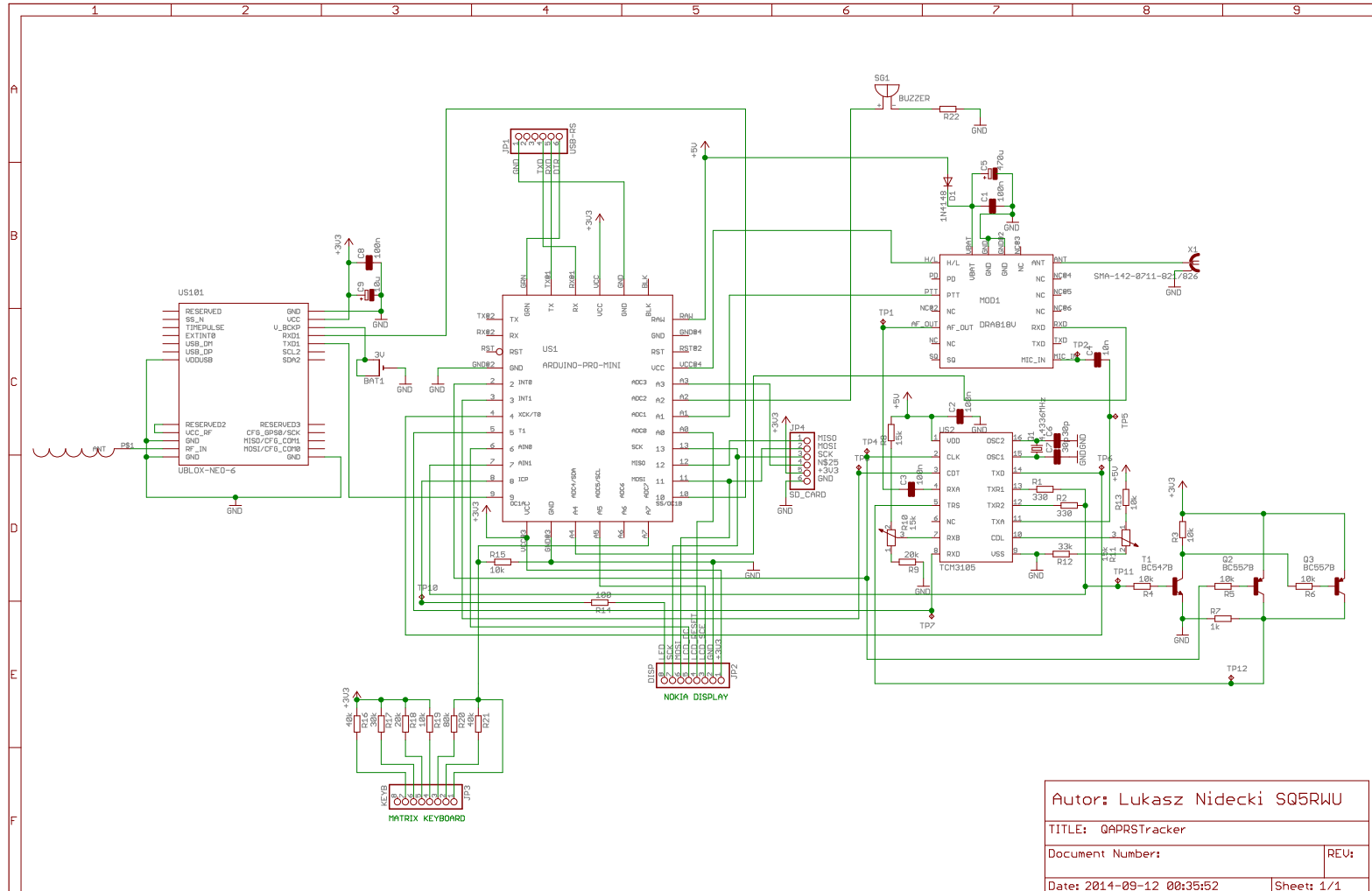
QAPRSTracker – opis urządzenia

QAPRSTracker jest urządzeniem pozwalającym na nadawanie i odbieraniem ramek w ramach sieci APRS. W jego skład wchodzi:

- gotowy moduł transcywera radiowego na pasmo 2m firmy DORJI, o mocy 0.5/1W,
- modem AFSK pod postacią układu TCM3105 (do kupienia na Ebay),
- wyświetlacz do NOKII 5110 (graficzny, b&w, 84x48px),
- odbiornik GPS NEO6-M firmy uBLOX,
- karta SD (jako pamięć stała),
- moduł Arduino Pro Mini 16 MHz z wymienionym stabilizatorem zasilania na 3.3V

Urządzenie pokazuje na wyświetlaczu odebrane i zdekodowane informacje z sieci APRS. Przy włączonym nadawaniu po ustaleniu pozycji z GPS następuje periodyczne wysyłanie ramek z użyciem SmartBeaconing.

Schemat



Zasada działania

Odbiór

Moduł **DRA818V MOD1** ustawiany jest programowo na odbiór transmisji FM na częstotliwości 144.800MHz.

Sygnał audio z wyjścia AF_OUT podawany jest poprzez kondensator odsprężający C3 na wejście RXA modemu **TCM3105 US2**. Potencjometr R10 - podłączony do wejścia RXB - pozwala na regulację biasu sygnału wejściowego co przekłada się na próg zadziałania detekcji tonów na wejściu RXA. Układ tranzystorów **T1** do **T3** wraz z rezystorami **R1** do **R7** pozwalają na podawanie na wejście TRS zanegowanego sygnału zegarowego z wyjścia CLK układu **US2** oraz sygnału na wejścia TXR1 i TXR2 – pozwala to na ustawienie odpowiedniej prędkości transmisji danych, oraz na przełączanie pomiędzy trybem nadawania i dekodowania danych przez modem.

Wykryte sygnały mark i space (czyli tony 1200 i 2200Hz) powodują zmianę stanu na wyjściu RXD, tak, że ton mark daje stan wysoki, a space niski. Sygnał ten doprowadzony jest do pinu 5 modułu Arduino (US1). Sygnał CLK doprowadzony do pinu 2 wyzwała przerwanie z częstotliwością 19,1kHz. W obsłudze przerwania sprawdzany jest stan pinu 5 i następuje dalsze dekodowanie pakietów.

Nadawanie

Moduł **DRA818V MOD1** ustawiany jest programowo na nadawanie transmisji FM na częstotliwości 144,800 MHz. Po przygotowaniu danych do wysłania następuje załączenie nadawania poprzez podanie stanu niskiego na wejście PTT w module. Wejście H/L modułu podłączone jest do napięcia zasilania mikrokontrolera, co wymusza użycie 1W. Podłączenie tego wejścia do masy zmniejsza moc do 0.5W (katalogowo).

Na wejście TXD układu **US2** podawane są kolejne bity do wysłania. Czas trwania poszczególnych sygnałów reguluje mikrokontroler poprzez odliczenie odpowiedniej ilości przerwań na pinie 2 **US1**. Modem na wyjściu TXA udostępnia sygnał audio (tony mark/space, w zależności od TXD) który poprzez kondensator **C4** podawany jest na wejście mikrofonowe MIC_IN **MOD1**.

Zasilanie

Układ zasilanie jest napięciem 5 V. Wymagana jest wydajność prądowa około 1A, ponieważ moduł transcevera może pobierać nawet 750mA przy nadawaniu. Nota katalogowa dla DRA818V przewiduje dla niego maksymalne napięcie zasilania na poziomie 4,5V. Dlatego też w szereg z zasilaniem włączona została dioda 1N4148 na której następuje spadek napięcia 0.5-0.6V (w zależności od pobieranego prądu). Dodatkowo kondensatory C1 i C5 filtrują zasilanie do radia.

Standardowo moduł Arduino Pro Mini zasilany jest napięciem 5 V. Na płytce ma on zamontowany stabilizator zasilania LDO 5V, do którego zasilanie doprowadzone jest na pin RAW. Został on w prototypie wymieniony na stabilizator 3,3 V które to napięcie niezbędne jest do zasilania modułu GPS, wyświetlacza i karty SD. Można również zrezygnować z przerabiania modułu, nie doprowadzać napięcia 5V do wejścia RAW i użyć dodatkowego stabilizatora 3.3V z którego napięcie należy doprowadzić do pinu VCC Arduino.

GPS

Jako odbiornik GPS użyty został układ [NEO6-M](#) firmy u-blox. Komunikacja z Arduino odbywa się za pomocą portu UART po stronie GPS, biblioteki SoftwareSerial po stronie mikrokontrolera. Praktyka wskazuje jednak na potrzebę użycia innej biblioteki do emulacji portu szeregowego albo stworzenia własnej implementacji – wbudowana biblioteka powoduje zakłócenia w dekodowaniu pakietów APRS poprzez naruszenie zależności czasowych w procedurach obsługi przerw.

Do odbiornika dołączona jest pastylkowa bateria 3 V podtrzymująca pamięć odbiornika po odłączeniu zasilania – znacznie przyspiesza to ustalanie pozycji po restarcie urządzenia.

Wyświetlacz

Do komunikacji z użytkownikiem zastosowany został graficzny wyświetlacz od Nokii 5110. Jest to popularny i tani wyświetlacz ciekłokrystaliczny, czarno-biały i rozdzielczości 84 na 48 pikseli z podświetleniem LED. Sterownik wyświetlacza to układ **PCD8544** – do jego obsługi w środowisku Arduino zastosowano gotową bibliotekę.

Karta SD

Jako że mikrokontroler Atmega328p ma małą ilość pamięci RAM, jak i EEPROM do zapisywania danych użyto karty SD o pojemności 64 MB. Pozwoli to na zapisanie ponad 60 tysięcy pakietów zanim nastąpi jej zapełnienie. Celem przyspieszenia operacji na danych nie użyto żadnego systemu plików na karcie – zamiast tego skorzystano z faktu, że karta SD zapisuje informacje w blokach po 512 bajtów i poszczególne pakiety zapisywane są w osobnych blokach.

Klawiatura

Jako klawiaturę zastosowany gotowy podzespół klawiatury matrycowej 3×4 klawisze. Została ona podłączona do układu rezystorów tak, że naciśnięcie klawisza tworzy dzielnik napięcia które jest następnie mierzone za pomocą wbudowanego przetwornika analogowo-cyfrowego. Celem obsługi różnych klawiatur i z uwagi na rozrzut wartości rezystorów przy pierwszym uruchomieniu układu następuje kalibracja klawiatury a dane kalibracyjne zapisywane są w pamięci EEPROM.

Konfiguracja

Konfiguracja urządzenia następuje poprzez port szeregowy. Należy podłączyć odpowiedni adapter UART<--> TTL do złącz Rx/Tx modułu Arduino. Można w tym celu wykorzystać np. popularne adaptory wpinane do portu USB. Ustawienia transmisji to bitrate 57600, 8 bitów danych, 1 bit stopu, bez kontroli parzystości. Aby przejść w tryb konfiguracji należy przy włączeniu zasilania wysłać po porcie szeregowym literę S. Po poprawnym rozpoznaniu urządzenie wyświetli na swoim wyświetlaczu napis SETUP. Oznacza to, że możemy przystąpić do konfigurowania go. Wszystkie komendy należy zakończyć znakiem nowej linii.

1. Wyświetlenie aktualnej konfiguracji **P**<enter>
2. Zapisanie ustawień w pamięci EEPROM **Q**<enter>
3. Zmiana:
 - Naszego znaku **SC**, np.: SCSQ5RWU<enter>
 - Naszego SSID **SS**, np. SS7<enter>
 - Tabeli symbolu graficznego **ST**, np. ST/<enter>
 - Symbolu graficznego **SY**, bp SY[<enter>
 - Ścieżki, z jaką nadajemy pakiety **SR**, np. SRWIDE1-1,WIDE2-1<enter> (max. 2 elementy!)
 - Komentarza **SO**, np. Sopożdrawiam Lukasz 73!<enter>

Oprogramowanie

Oprogramowanie urządzenia bazuje na bibliotece ArduinoQAPRS – w wersji z obsługą dekodowania danych nazywanej roboczo ArduinoQAPRSPlus. Kod źródłowy zarówno zmodyfikowanej biblioteki jak i QAPRSTrackera zostanie udostępniony na licencji GNU.