

projekt

# „radio3”

autor: Robert Jaremczak, SQ6DGT

wersja 1.1 – robocza

aktualizacja: 2017-03-26

# 1 Protokół komunikacyjny

Komunikacja z urządzeniem polega na wymianie ramek w trybie master/slave przez port szeregowy, emulowany za pomocą interfejsu USB. Urządzenie pracuje w trybie slave i wysyła ramki tylko w odpowiedzi na żądania pochodzące z zewnątrz, samodzielnie nie inicjuje transmisji.

Standardowe parametry transmisji szeregowej:

- prędkość: 115200 bit/s
- bity danych: 8
- bity stopu: 1
- kontrola parzystości: wyłączona

## 1.1 Struktura ramki

Ramka ma następującą strukturę:

długość [B]	nazwa	opis
2	header	format ramki i kod komendy
0..2	payload_length	wartość określająca ilość danych do przesłania
0..65804	payload	dane do przesłania
1	crc8	liczba kontrolna wyliczona algorytmem CRC8

Struktura nagłówka jest następująca:

<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
frame_type				fame_command											

Maksymalna ilość danych, która może zostać przesłana w pojedynczej ramce to 65804 bajty. Wartości i długości odpowiednich pól ramki zależą od jej typu, który z kolei zależy od ilości danych do przesłania.

frame_type	frame_format	payload_length	payload
A	ilość danych	brak	0 – 13 bajtów
B	14	ilość danych – 14	14 – 269 bajtów
C	15	ilość danych – 270	270 – 65804 bajtów

Ten sposób konstruowania ramki ma na celu zminimalizowanie narzutu w przypadku krótkich komunikatów. Pole crc8 jest wyliczane dla wszystkich bajtów ramki włącznie z nagłówkiem.

## 2 Zaimplementowane żądania

Program sterujący wysyła ramkę z żądaniem do urządzenia, które odsyła odpowiedź. Jeżeli ramka nie zawiera dodatkowych danych w opisie występuje tylko kod komendy. Jeśli dane występują są opisane w formacie nazwa\_pola[długość w bajtach]. Każdorazowo format ramki jest odpowiednio dobrany w zależności od całkowitej ilości danych do przesłania.

### 2.1 PING

test połączenia, komunikat potwierdzający otrzymanie żądania.

żądanie	
frame_command	0x000
odpowiedź	
frame_command	0x000

### 2.2 DEVICE\_INFO

szczegółowe dane o urządzeniu

żądanie	
frame_command	0x001
odpowiedź	
frame_command	0x001
name[16]	char[]: nazwa urządzenia
buildId[32]	char[]: tekstowy identyfikator wersji oprogramowania
hardwareRevision[1]	uint8_t: wersja sprzętu: 0 - wersja 1 i wcześniejsze 1 - wersja 2
vfoType[1]	uint8_t: rodzaj zainstalowanego VFO: 0 - brak VFO 1 - DDS AD9850 2 - DDS AD9851
baudRate	uint32_t: efektywna prędkość transmisji

### 2.3 DEVICE\_STATE

aktualny stan urządzenia

żądanie	
frame_command	0x002
odpowiedź	
frame_command	0x002
timeMs[4]	uint32_t: liczba milisekund od włączenia zasilania
analyser[1]	uint8_t: stan analizatora: 0 - gotowy do obsługi żądań 1 - przetwarza żądanie 2 - niewłaściwe żądanie, oczekuje na następne
vfoOut[1]	bool: przekierowanie wyjścia VFO

	0 - bezpośrednio na zewnętrzne wyjście VFO 1 - na wejście modułu VNA
vfoAmplifier[1]	bool: wzmacniacz sygnału VFO włączony wymagana wersja sprzętu: 2
vfoAttenuator[1]	uint8_t: sterowanie sekcjami tłumika VFO 0..7 - każdy bit załącza odpowiednią sekcję tłumika wymagana wersja sprzętu: 2
logLevel[1]	uint8_t: szczegółowość log-u od strony urządzenia 0 - debug 1 - info 2 - error

## 2.4 VFO\_GET\_FREQ

pobierz aktualną częstotliwość VFO

żądanie	
frame_command	0x008
odpowiedź	
frame_command	0x008
frequency[4]	uint32_t: częstotliwość w Hz

## 2.5 VFO\_SET\_FREQ

ustawienie częstotliwości VFO

żądanie	
frame_command	0x009
frequency[4]	uint32_t: częstotliwość w Hz
odpowiedź	
patrz: PING	

## 2.6 LOGPROBE\_GET

Odczyt przetwornika A/C sondy logarytmicznej

żądanie	
frame_command	0x010
odpowiedź	
frame_command	0x010
value[2]	uint16_t: uśredniona wartość z przetwornika A/C

## 2.7 LINPROBE\_GET

Odczyt przetwornika A/C sondy liniowej

żądanie	
frame_command	0x018
odpowiedź	
frame_command	0x018
value[2]	uint16_t: uśredniona wartość z przetwornika A/C

## 2.8 CMPPROBE\_GET

Odczyt przetworników A/C z komparatora modułu VNA

żądanie	
frame_command	0x020
odpowiedź	
frame_command	0x020
gain[2]	uint16_t: uśredniona wartość z przetwornika A/C
phase[2]	uint16_t: uśredniona wartość z przetwornika A/C

## 2.9 FMETER\_GET

Odczyt z licznika częstotliwości

żądanie	
frame_command	0x028
odpowiedź	
frame_command	0x028
frequency[4]	uint32_t: liczba impulsów zliczona w ciągu sekundy

## 2.10 PROBES\_GET

Jednoczesny odczyt wartości A/C wszystkich sond

żądanie	
frame_command	0x030
odpowiedź	
frame_command	0x030
logarithmic[2]	uint16_t: uśredniona wartość A/C sondy logarytmicznej
linear[2]	uint16_t: uśredniona wartość A/C sondy liniowej
vna_gain[2]	uint16_t: uśredniona wartość A/C wyjścia „gain”
vna_phase[2]	uint16_t: uśredniona wartość A/C wyjścia „phase”
frequency[4]	uint32_t: liczba impulsów zliczona w ciągu sekundy

## 2.11 VFO\_OUT\_DIRECT

Przekierowanie wewnętrznego VFO bezpośrednio na wyjście VFO

żądanie	
frame_command	0x033
odpowiedź	
patrz: PING	

## 2.12 VFO\_OUT\_VNA

Przekierowanie wewnętrznego VFO na wejścia modułu VNA

żądanie	
frame_command	0x034
odpowiedź	

patrz: PING

## 2.13 ANALYSER\_REQUEST

Wykonanie analizy z przemiataniem częstotliwości. Żądanie jest ignorowane jeśli poprzednie nie zostało jeszcze zakończone. W każdym przypadku zwracana jest odpowiedź konkludująca zawierająca status (DEVICE\_STATE).

żądanie	
frame_command	<b>0x040 (ANALYSER_REQUEST)</b>
freqStart[4]	uint32_t: częstotliwość początkowa
freqStep[4]	uint32_t: krok częstotliwości
numSteps[2]	uint16_t: liczba kroków do wykonania
source[1]	uint8_t: źródło danych wejściowych 0 - sonda logarytmiczna 1 - sonda liniowa 2 - komparator VNA, wyjścia „gain” i „phase”
odpowiedź	
frame_command	<b>0x042 (ANALYSER_DATA)</b>
analyserState[1]	uint8_t: stan końcowy operacji przemiatania 0 - wykonano poprawnie 1 - przetwarzanie trwa 2 - żądanie nieprawidłowe
freqStart[4]	uint32_t: częstotliwość początkowa wykonanej analizy
freqStep[4]	uint32_t: krok częstotliwości
numSteps[2]	uint16_t: liczba wykonanych kroków
source[1]	uint8_t: źródło danych wejściowych (patrz wyżej)
data[0-4004]	uint16_t[]: tablica danych pobranych z wybranego źródła. Pojedynczy punkt pomiarowy zawiera 1 lub 2 słowa 16 bitowe z wartością/wartościami pobranymi z wybranego źródła danych pomiarowych. Maksymalna liczba kroków pomiarowych wynosi 1001 co daje maksymalnie 4004 bajty dla źródła: 2 (komparator VNA)

## 2.14 **DEVICE\_HARDWARE\_REVISION**

Ustawienie wersji sprzętu lub uruchomienie automatycznej detekcji

żądanie	
frame_command	<b>0x003</b>
hardwareRevision[1]	uint8_t: określenie żądanej wersji sprzętu 0 - uruchomienie mechanizmu detekcji automatycznej 1 - wersja 1 i wcześniejsze 2 - wersja 2
odpowiedź	
patrz: DEVICE_INFO	

## 2.15 **VFO\_TYPE**

Określenie zainstalowanego typu VFO

żądanie	
frame_command	<b>0x035</b>
vfoType[1]	uint8_t: rodzaj zainstalowanego VFO 0 - brak 1 - DDS z układem AD9850, moduł „gotronik” 2 - DDS z układem AD9851, moduł „gotronik”
odpowiedź	
patrz: DEVICE_INFO	

## 2.16 **VFO\_ATTENUATOR**

Ustawienie tłumika sygnału VFO

żądanie	
frame_command	<b>0x036</b>
vfoAttenuator[1]	uint8_t: 0..7 poziom tłumienia wymagana wersja sprzętu: 2
odpowiedź	
patrz: DEVICE_STATE	

## 2.17 **VFO\_AMPLIFIER**

Sterowanie wzmacniaczem sygnału VFO

żądanie	
frame_command	<b>0x037</b>
vfoAmplifier[1]	bool: włączenie wzmacniacza sygnału VFO wymagana wersja sprzętu: 2
odpowiedź	
patrz: DEVICE_STATE	

## 2.18 **VNA\_MODE**

Przełączanie trybu pracy modułu analizatora VNA

żądanie
---------

frame_command	<b>0x038</b>
vnaMode[1]	uint8_t: tryb pracy 0 - sprzęgacz kierunkowy i komparator 1 - mostek pomiarowy i komparator wymagana wersja sprzętu: 2 wymagana wersja modułu VNA: 2
odpowiedź	
patrz: DEVICE_STATE	

### 3 Typowe sekwencje komend

#### 3.1 Sekwencja startowa

Sekwencja startowa powinna zostać wykonana zaraz po otwarciu portu szeregowego służącego do komunikacji z urządzeniem.

- wysłanie DEVICE\_HARDWARE\_REVISION
- wysłanie VFO\_TYPE
- pobranie informacji o urządzeniu: DEVICE\_INFO
- pobranie stanu urządzenia: DEVICE\_STATE

jeśli wszystkie zwrócone dane są prawidłowe można rozpocząć pracę.