

projekt

„radio3”

autor: Robert Jaremczak, SQ6DGT

wersja: 1.1 – robocza

aktualizacja: 2017-04-09

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 zawiera sekwencję pól:

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

z których nie każde musi wystąpić. Zależy to od ilości dodatkowych danych, które mają być przesłane. W każdym przypadku ramka rozpoczyna się nagłówkiem, którego struktura jest następująca:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
frame_type				frame_command											

Kolejnym elementem występującym zawsze jest pole **frame_command** czyli rodzaj żądania. Maksymalna ilość danych, która może zostać przesłana w pojedynczej ramce to 65804 bajty. W zależności od ilości danych do przesłania wybierany jest odpowiedni rodzaj ramki.

frame_type	payload_length (ilość danych)	payload_content (dane)
ilość danych	brak	0 – 13 bajtów
14	ilość danych – 14	14 – 269 bajtów
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 za pomocą standardowego wielomianu $X^8+X^5+X^4+X^0$. Pola o długości większej niż jeden bajt są konstruowane w konwencji *little-endian*, czyli młodszy bajt jest wysyłany jako pierwszy.

1.2 przykład 1: ramka PING

frame_type: 0 - brak danych do przesłania

frame_command: 0 - żądanie PING

payload_length: to pole nie występuje

payload_content: to pole nie występuje

wysłane bajty: **00 00 CRC8** (3 bajty)

1.3 przykład 2: ramka DEVICE_HARDWARE_REVISION

frame_type: 0x1 – ilość dodatkowych bajtów wynosi 1

frame_command: 0x003 – taki jest kod żądania

payload_length: to pole nie występuje, długość zakodowana w frame_type

payload_content: jeden bajt określający wersję sprzętu (np. 2)

wysłane bajty: **10 03 02 CRC8** (4 bajty)

1.4 przykład 3: ramka SWEEP_RESPONSE

Zakładamy, że przesłano 1000 próbek (2 bajty na próbkę)

frame_type: 0xF – ponieważ ilość danych przekracza 270 bajtów

frame_command: 0x041 – kod żądania

payload_length: 2 bajty długości zawierające liczbę: $2014 - 270 = \mathbf{1744}$

payload_content: 12 bajtów parametrów + 2002 bajty próbek

wysłane bajty: **F0 41 D0 06 ... CRC8** (2019 bajtów)

1.5 wyliczanie pola kontrolnego wg CRC8

Wykorzystany jest standardowy algorytm CRC8 z domyślnym wielomianem $X^8+X^5+X^4+X^0$. Przykładowe obliczenie:

bajty: **1A 1B 2F FF 01 23**

crc8: **A5**

kalkulator CRC8: http://tomeko.net/online_tools/crc8.php?lang=pl

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
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

2.4 DEVICE_HARDWARE_REVISION

Ustawienie wersji sprzętu lub uruchomienie automatycznej detekcji

żądanie	
frame_command	0x003
hardwareRevision[1]	uint8_t: określenie żądanej wersji sprzętu 0 - uruchomienie mechanizmu detekcji automatycznej 1 - wersja 1 i wcześniejsze

	2 - wersja 2
	odpowiedź
PING	

2.5 VFO_GET_FREQ

pobierz aktualną częstotliwość VFO

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

2.6 VFO_SET_FREQ

ustawienie częstotliwości VFO

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

2.7 LOGPROBE_DATA

Odczyt przetwornika A/C sondy logarytmicznej

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

2.8 LINPROBE_DATA

Odczyt przetwornika A/C sondy liniowej

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

2.9 VNAPROBE_DATA

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

2.10 FMETER_DATA

Odczyt z licznika częstotliwości

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

2.11 PROBES_DATA

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

2.12 VFO_OUT_DIRECT

Przekierowanie wewnętrznego VFO bezpośrednio na wyjście VFO

żądanie	
frame_command	0x033
odpowiedź	
PING	

2.13 VFO_OUT_VNA

Przekierowanie wewnętrznego VFO na wejścia modułu VNA

żądanie	
frame_command	0x034
odpowiedź	
PING	

2.14 SWEEP_REQUEST

Wykonanie analizy z przemiataniem częstotliwości. Żądanie jest ignorowane jeśli poprzednie nie zostało jeszcze zakończone.

żądanie	
---------	--

frame_command	0x040 (SWEEP_REQUEST)
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”
avg.mode[1]	uint8_t: tryb uśredniania próbek i cykli przemiatania bity 0-3 - liczba próbek do uśredniania - 1 bity 4-7 - liczba cykli przemiatania - 1
odpowiedź	
frame_command	0x041 (SWEEP_RESPONSE)
state[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 (0 - w razie błędu)
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)

2.15 VFO_TYPE

Określenie zainstalowanego typu VFO

żądanie	
frame_command	0x035
vfoType[1]	uint8_t: rodzaj zainstalowanego VFO 0 - brak 1 - DDS z układem AD9850, moduł „gotronik” 2 - DDS z układem AD9851, moduł „gotronik”
odpowiedź	
PING	

2.16 VFO_ATTENUATOR

Ustawienie tłumika sygnału VFO

żądanie	
frame_command	0x036
vfoAttenuator[1]	uint8_t: 0..7 poziom tłumienia wymagana wersja sprzętu: 2
odpowiedź	
PING	

2.17 VFO_AMPLIFIER

Sterowanie wzmacniaczem sygnału VFO

żądanie	
frame_command	0x037
vfoAmplifier[1]	bool: włączenie wzmacniacza sygnału VFO wymagana wersja sprzętu: 2
odpowiedź	
PING	

2.18 VNA_MODE

Przełączanie trybu pracy modułu analizatora VNA

żądanie	
frame_command	0x038
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	
PING	

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.