

# Moduł precyzyjnego generatora TCXO i synchronizacji z GPS

część 2

W EdW 9/2020 pojawił się artykuł opisujący licznik częstotliwości i czasu AVT3275. Czymże on byłby, gdyby nie miał precyzyjnego generatora? Byłby miernikiem z dużą liczbą cyfr, z których połowa nie miałaby żadnego znaczenia!



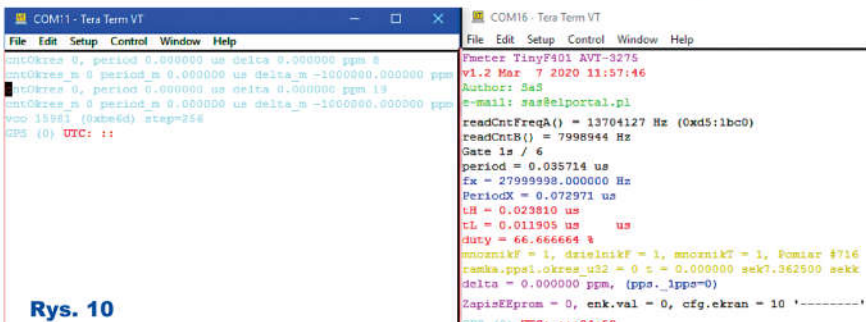
Pierwsza część artykułu zakończyła się opisem wstępnego uruchomienia. Oto dalsze wskazówki.

W następnej kolejności trzeba przyłączyć konwerter USB-UART do J17. W programie terminalu należy ustawić wymagany port COM z parametrami 921600 8N1. Na ekranie powinniśmy zobaczyć obraz jak na **rysunku 10**. Prawe okno terminalu przedstawia informacje wysyłane przez AVT3275 na J7, gdy jest on połączony z generatorem przez J11. O sposobie podłączenia częstościomierza można przeczytać w dalszej części artykułu. Jeśli generator będzie synchronizowany sygnałem z GPS, należy go przyłączyć do J4. Do wyprowadzenia 4 przyłączamy linię TX modułu GPS, która jest linią RX UART w mikrokontrolerze.

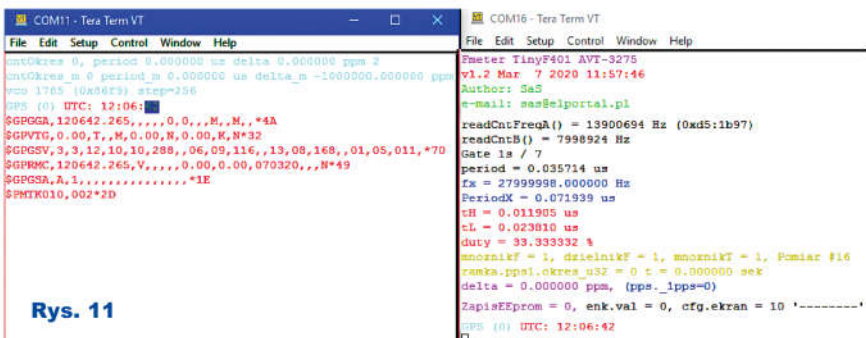
RX GPS można podłączyć do wyprowadzenia 1, ale na razie nie jest ona używana. Do wyprowadzenia 3 doprowadzamy sygnał 1pps. **Fotografia 4** pokazuje miejsce, w którym ten sygnał jest dostępny dla modułu GY-GPS6MV2.

Dla modułu FGPM-MOSL3C będzie to wyprowadzenie 9 i tego właśnie modułu użyto w prototypie. Na wyprowadzeniu 5 J4 dostępny jest sygnał reset. Gdy komunikacja z modułem przebiega poprawnie, w terminalu zobaczymy ekran podobny do tego z **rysunku 11**, gdy GPS nie

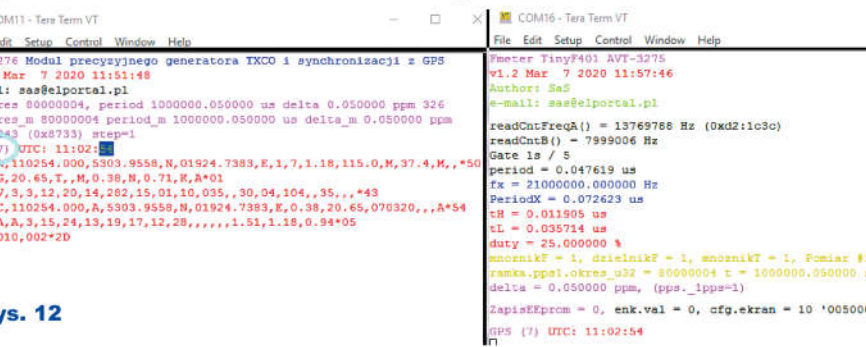
odbiera sygnału z satelitów. W chwili odebrania sygnału z satelitów wygląd ekranu zmieni się na podobny do **rysunku 12**,



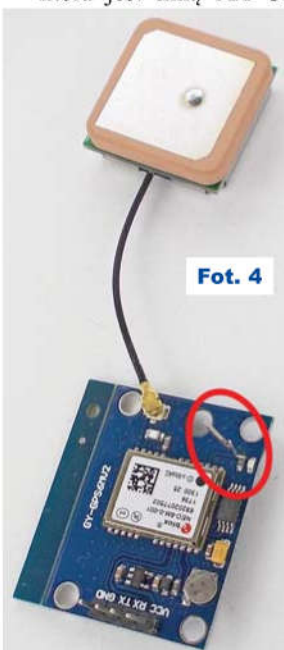
Rys. 10



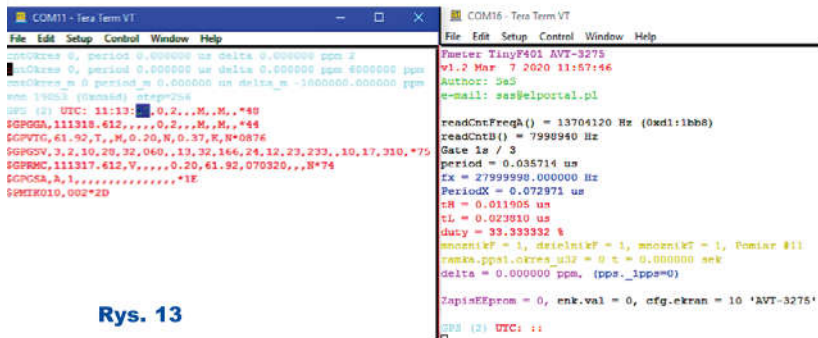
Rys. 11



Rys. 12



Fot. 4



Rys. 13



Fot. 5

gdzie widać odbiór z siedmiu satelitów. Jeśli moduł GPS nie dostarcza sygnału wzorcowego 1pps, ekran będzie wyglądał podobnie jak na rysunku 13. Informacje te są także wyświetlane na wyświetlaczu OLED – fotografia 5. W górnym wierszu wyświetlany jest czas UTC, następnie w nawiasie liczba satelitów, po niej czas ostatniej synchronizacji. W dolnym wierszu odchyłka TCXO od sygnału 1pps z GPS.

Gdy moduł jest uruchomiony, można podłączyć go do częstotściomierza. W przypadku AVT3275 należy wylutować kwarce i do wejścia X1 przylutować przewód lub goldpin (fotografia 6) oraz doprowadzić sygnał 8 MHz, który jest dostępny na wyprowadzeniu 4 złącza J16. Połączenie to powinno być wykonane kablem ekranowanym. W prototypie połączenie nieekranowanym przewodem długości 15 centymetrów powodowało zakłócenia obrazu pobliskiego telewizora. Moduł można zasilić z płytki AVT3275, łącząc wyprowadzenie 1 J11 częstotściomierza z 1 J16 (+3,3V) generatora i 2 J11 z 2 J16 (GND). Dodatkowo wyprowadzenie 3 J11 warto połączyć z linią TX UART dostępną na wyprowadzeniu 4 J15 generatora. Dzięki temu na ekranie 10 częstotściomierza będzie można odczytać informacje o aktualnej dokładności generatora, bieżącym czasie UTC, liczbie satelitów, z których GPS odbiera sygnał oraz czasie ostatniej synchronizacji TCXO z GPS – fotografia 7. Aby ta funkcjonalność działała, należy uaktualnić program w AVT3275 do wersji 1.2 lub nowszej. Program dostępny jest na Elportalu. Połączony zestaw AVT3275 z AVT3276 można zobaczyć na fotografii 8.

Nie opisuję sposobu podłączenia generatora do AVT5575, bo używanie

go z tym częstotściomierzem nie ma sensu.

Generator co sekundę wysyła informacje o parametrach pracy. Aby wykorzystać te informacje, trzeba znać format ramki wysyłanej przez UART. Dane są wysyłane z prędkością 9600 w formacie 8N1 w postaci ramki składającej się z nagłówka i chunków:

```
struct{
    header_t header;
    chunk_nr nr;
    chunk_lpps lpps;
    chunk_delta delta;
    chunk_gps gps;
    chunk_end end;
}ramka_t;
nagłówek:
typedef struct{
    uint32_t crc;
    uint16_t len;
    uint32_t id;
    uint16_t ver;
}header_t;
```

zawiera sumę kontrolną CRC obejmującą całą strukturę „ramka\_t” bez CRC. Następna jest długość ramki równa wszystkim danym, łącznie z CRC, po tym jest ID programu o wartości 0x636e7953, następnie wersja programu. Wszystkie chunki mają taką samą budowę, na którą składa się:

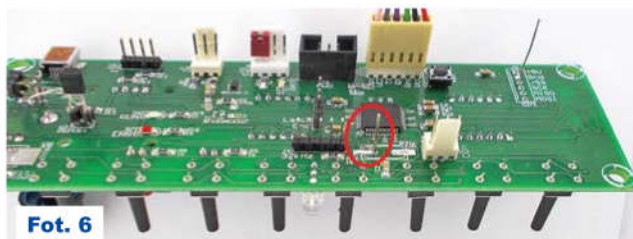
ID	UINT8	unikalny identyfikator
LEN	UINT16	długość chunka w bajtach
DATA	UINT8*	dane o długości LED. Gdy LEN=0 danych nie ma

Pierwszy chunk:

```
typedef struct{
    uint8_t id;
    uint8_t len;
    uint32_t cnt;
}chunk_nr;
```

o id=1, len=4 zawiera tylko pole „cnt”, w którym jest zawarty numer kolejnej transmisji. Następny chunk:

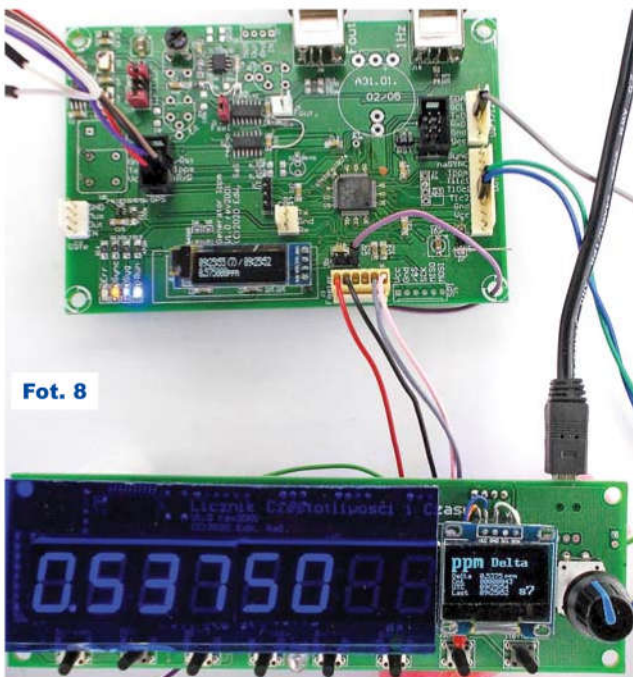
```
typedef struct{
    uint8_t id;
    uint8_t len;
```



Fot. 6



Fot. 7



Fot. 8

```
uint32_t okres_u32;
double okres_d;
float okres_f;
}chunk_lpps;
```

id=2, len=16, „okres\_u32” zawiera liczbę zliczonych impulsów zegarowych (80MHz) w czasie jednosekundowego pomiaru okresu sygnału wzorcowego. Przy idealnej synchronizacji będzie to 80e6. Zmienna „okres\_d” zawiera zmierzony okres sygnału przedstawiony w postaci liczby zmiennoprzecinkowej double, „okres\_f” to samo w formacie float. Kolejny chunk:

```
typedef struct{
    uint8_t id;
    uint8_t len;
    double delta_d;
    float delta_f;
}chunk_delta;
```

id=3, len=12, „delta\_d” to odchyłka od okresu jednej sekundy typ double, „delta\_f” to samo typ float. Po nim chunk:

```
typedef struct{
  uint8_t id;
  uint8_t len;
  char utc[6];
  char lastUTC[6];
  uint8_t satelit;
}chunk_gps;
chunk_gps;
id=4, len=13, „utc” zawiera w formie
tekstu godzinę w formacie ggmms,
„lastUTC” – czas ostatniej synchroniza-
cji z GPS. Ostatni chunk:
typedef struct{
  uint8_t id;
  uint8_t len;
}chunk_end;
nie zawiera danych, id=0, len=0.
```

Trzeba mieć świadomość, że w kolej-  
nych wersjach oprogramowania, rozmiar  
chunków może się zmienić (dojdą nowe  
informacje), tak samo ich kolejność nie  
jest gwarantowana poza chunkami „hea-  
der” i „end”. Całości dopełnia definicja  
identyfikatorów chunków

```
enum{ CHUNK_END, CHUNK_NR_TRANSMISJI, CHUNK_IPPS, CHUNK_DELTA, CHUNK_GPS };
```

oraz funkcja obliczająca CRC i sposób  
jej użycia:

```
uint32_t Crc( u8 *p, u16 len ){
  return CalculateCrc32( p, len, len );
}

p = (uint8_t*)&ramka; // Wskaźnik na strukturę
p += sizeof(ramka.header.crc); // Zwiększamy o rozmiar CRC
uint16_t len = ramka.header.len;
uint32_t crc = Crc( p, len );
```

W przypadku wykorzystania generatora  
z innymi urządzeniami do komunikacji  
można też wykorzystać interfejs I2C.  
Aby działał poprawnie, należy założyć  
zworkę na J9. Interfejs ten jest dostępny  
także na AVT3275, ale na razie I2C  
w generatorze może pracować z maksy-  
malnym taktowaniem 100kHz. W czę-  
stościomierzu do magistrali I2C przyłą-  
czony jest wyświetlacz OLED, który

korzysta z trybu FAST  
(400kHz). Zmniejszenie  
szybkości komunikacji  
z wyświetlaczem powodo-  
wało widoczne odświeżanie  
zawartości ekranu, dlatego  
zrezygnowano z komunikacji  
z generatorem przez I2C.  
Taka komunikacja będzie  
możliwa dla rozbudowanej  
wersji częstościomierza,  
który będzie miał kilka inter-  
fejsów I2C ale niewykluczo-  
ne, że prędkość komunikacji  
w generatorze zwiększy się  
i możliwa będzie bezpośrednia  
współpraca częstościomie-  
rza z generatorem przez  
I2C. Opcjonalnym rozwiąza-  
niem będzie płytka obwodowa

wejści-  
owych, na  
której znajdować się będzie multiplexer  
I2C PCA9544. Rozdzielał on będzie  
interfejs I2C na cztery  
magistrale. Każda z nich  
może pracować z inną  
prędkością. Do jednej  
z nich można będzie  
podłączyć płytkę gen-  
eratora. Komunikacja  
z generatorem przez I2C  
polega na zaadresowaniu rejestru oraz  
jego odczycie. Adres I2C generatora:  
0x38 (0x70). Zawartość poszczególnych  
rejestrów prezentuje **tabela 1**.

Prosty program dla Arduino dostępny  
na Elportalu pokazuje sposób komu-  
nikacji, a na **rysunku 14** widać efekt  
jego pracy. Jeśli master I2C obsługuje  
tryb *multimaster*, to kilka urządzeń może  
komunikować się z generatorem.

Gdy urządze-  
nie, które korzysta  
z generatora, nie ma  
interfejsu I2C lub  
UART, przydatne  
mogą być sygnały  
SYNC, NOSYNC  
i IPPM, dostępne  
na J16. Sygnał  
IPPM jest aktyw-  
ny, gdy na wejściu  
3 J4 dostępny jest  
sygnał wzorcowy,  
NOSYNC sygnali-  
zuje obecność syg-  
nału o dokładności  
powyżej 1ppm,  
SYNC 1ppm lub  
mniej (**tabela 2**).

**Zmiany w progra-  
mieczęstościomierza**

Rejestr	Offset	Nazwa	Typ danej	Zawartość
0	0	ID	uint32	0x636e7953
	4	NAME	char*	AVT-3276
	14	EMAIL	char*	sas@elportal.pl
	30	DATE	char*	data kompilacji programu
	37	TIME	char*	czas kompilacji
1	0	SAT	uint8	liczba satelitów, z których odbierany jest sygnał GPS
	1	UTC	char*	czas UTC z GPS
	7	lastUTC	char*	czas ostatniej synchronizacji z GPS
2	0	period	uint32	zawartość licznika zliczającego okres sygnału wzorcowego
3	0	period	float	okres sygnału wzorcowego
4	0	period	double	okres sygnału wzorcowego
5	0	delta	float	odchyłka TCXO od sygnału wzorcowego w ppm
6	0	delta	double	odchyłka TCXO od sygnału wzorcowego w ppm

**Tabela 1**

Symbol	Nazwa	Funkcja
D4	Run	Miganiem informuje o pracy programu.
D3	Err	Świeceniem lub miganiem informuje o błędzie.
D1	Syg	Świeceniem informuje o obecności sygnału wzorcowego 1Hz.
D2	Sync	Świeceniem informuje o odbiorze sygnału GPS.

**Tabela 2**

**AVT3275.**  
Z m i a n y  
d o t y c z ą  
wyświetla-  
nia infor-  
m a c j i

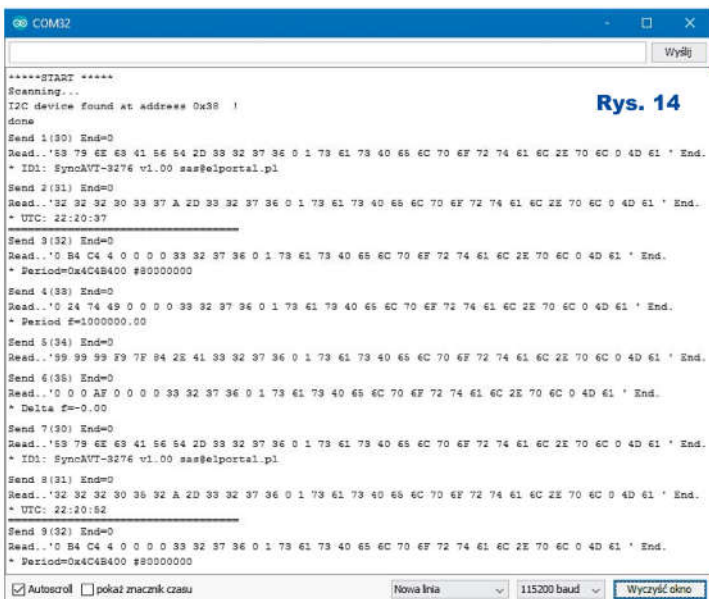


**Fot. 9**

na ekranie numer 10. Aktualnie  
wyświetlane są według **fotografii 9**.  
Delta – Odchyłka TCXO od wzorca 1Hz  
w ppm.  
CNT – Liczba impulsów zliczona przez  
licznik taktowany 80MHz mierzący  
okres sygnału 1Hz.  
UTC – Aktualny czas z GPS.  
Last – Czas ostatniej synchronizacji.

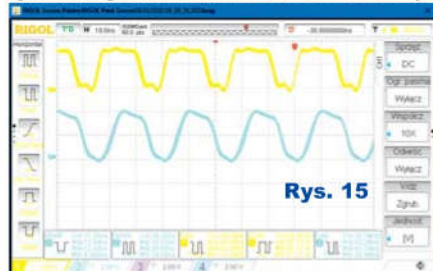
Rodzaj informacji wyświetlanej na  
LED wybiera się, naciskając przycisk  
enkodera.

**Słowo końcowe.** Obwód z U3B może  
posłużyć do prób i pomiaru różnych  
przebiegów. Korzystając z drugiego  
komparatora (złącze J2), przeprowadzo-  
no próby pomiaru częstotliwości gene-  
ratora 40MHz **I533-2P3-40.0000M**. Po  
ustawieniu potencjometrem P2 wypeł-  
nienia blisko 50%, sygnał wyglądał jak  
na **rysunku 15**. Kanał 1 (lazurowy)  
pokazuje wyjście komparatora, kanał 2  
(zółty) wyjście bramki Schmitta U4B.  
Jak widać, daleko im do „książkowe-  
go” prostokąta. Przyłączenie przewodu  
łączącego komparator z częstościomie-  
rzem AVT3275 silnie tłumy sygnał (**rys-  
nek 16**), natomiast gdy przewód przy-

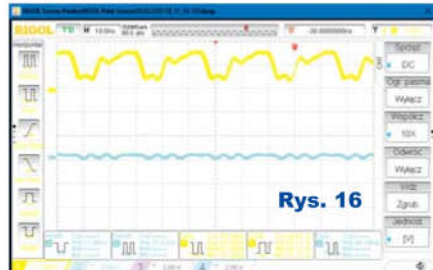


**Rys. 14**

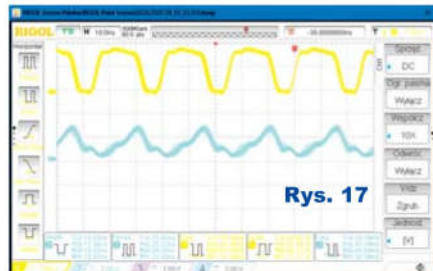
łączony jest wyjścia U4B, oscylogram wygląda jak na **rysunku 17**. Kolejne próby polegały na zmierzeniu częstotliwości generatora przez częstotliciemierz. W pierwszej kolejności AVT3275 podłączony był do wyjścia komparatora. Po próbach ustawienia potencjometru P2 udało się przy sygnale jak na oscylogramie z **rysunku 18** uzyskać poprawny wynik na wyświetlaczu LED i w oknie terminalu – **rysunek 19**. Warto zauważyć, że poprawny pomiar był możliwy przy takim ustawieniu, przy którym na wyjściu bramki Schmitta nie było sygnału! Niewielka zmiana ustawienia potencjometru powodowała niestabilne odczy-



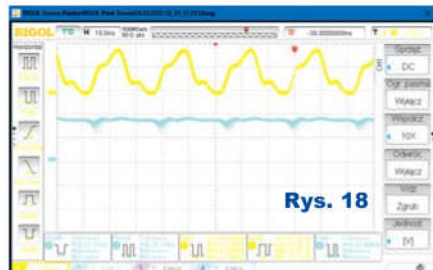
Rys. 15



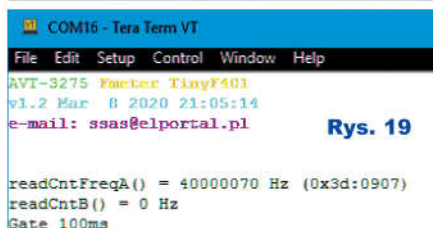
Rys. 16



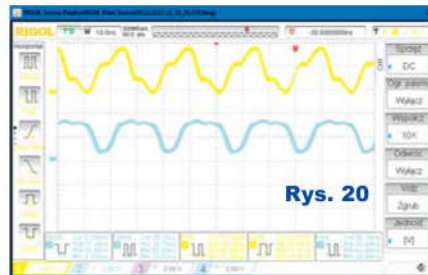
Rys. 17



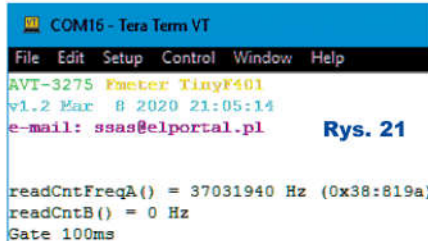
Rys. 18



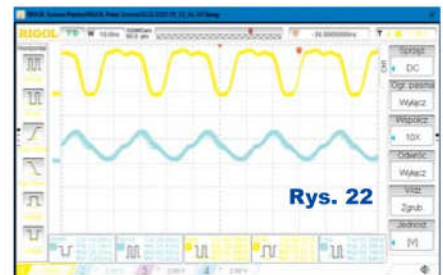
Rys. 19



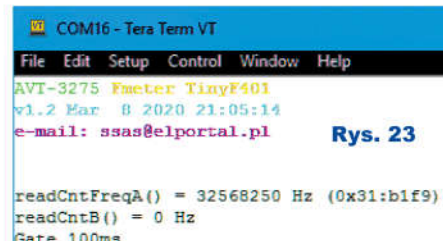
Rys. 20



Rys. 21



Rys. 22



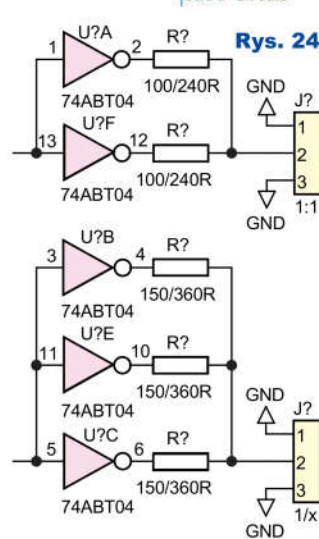
Rys. 23

ty: **rysunek 20** oraz **rysunek 21**. Próby pomiaru sygnału z wyjścia bramki U4B skończyły się niepowodzeniem: **rysunki 22** i **23**.

Próby takie uświadamiają, że budowa stopnia wejściowego łatwa nie będzie, a wydaje się, że to tylko kilka układów scalonych. Ułatwieniem będzie to, że połączenia pomiędzy układami będą krótkie, więc cechować je będzie mała pojemność. Tam, gdzie sygnał będzie przesyłany na większe odległości, zastosowane będą „dziwne” rozwiązania, jak pokazuje **rysunek 24**.

Dlatego do AVT3276 wybrano STM32F401?

Pierwotnie miał to być STM32F411RE pracujący z zegarem 100MHz. Po zasta-



Rys. 24

nowieniu wybrano taki sam mikrokontroler jak w licznikach AVT3275 i AVT5575, pracujący z częstotliwością 80MHz. Dzięki temu duża część kodu mogła zostać przeniesiona z liczników częstotliwości, co znacznie skróciło czas poświęcony na napisanie programu. Co prawda timer mierzący wzorec liczy „tylko” 80 milionów impulsów w ciągu sekundy (AVR liczyłby dziesięć razy mniej, dokładnie 8 milionów impulsów, chyba że wersja mikrokontrolera w małej obudowie, wtedy osiem razy mniej dokładnie, 10e6 impulsów).

SaS  
sas@elportal.pl

## Wykaz elementów

Rezystory 1206	
R1	10kΩ
R2	1kΩ
R5	1MΩ
R6,R12,R13,R14,R15	470Ω
R7	51Ω
R9,R10	100Ω
R3,R4,R8,R11	100Ω
R16,R17,R18,R19,R20	2,2Ω
P1,P2,P3	10kΩ pot. montażowy
Kondensatory 1206	
C1-C6,C8-C11,C13-C18,C21,C24	100nF
C7	1nF
C12	1μF
C19	2μF
C22	5,6pF
C23	5,6pF
C25	10μF
C26	10μF

## Półprzewodniki

U1	MCP6232 SO-8
U2	TG-5006CG-42L 32M 2ppm
U3	TLV3502 SO-08
U4	74HC14D SO-14
U5	opcja DIP8/14
U6	ASVTX-09-20M 0.5ppm *
U7	74HC74D SO-14
U8	STM32F401RCT6 TQFP-68
U9	74LVC1G157GW SC-88
U11	23K256-I/SN * SO-8
U10	24LC512 SO-8
U12a	DS2431+ *
U12b	DS2431P+ *
D1	Dioda LED zielona 1206
D2	Dioda LED żółta 1206
D3	Dioda LED czerwona 1206
D4	Dioda LED niebieska 1206
D5	BAS85 *

D7	SM4007
LCD1	Moduł modOLED091-IIC
Inne	
Q1	32768kHz *
BT1	CR2032-U *
J1	Goldpin 3x1
J2	NS25-W4P
J3	NS25-W6P
J4	NS25-W3P
J6	Gniazdo BNC
J5	NS25-W2P
J4	T821-1-06-S1
J16	NS25-W8P
JP2	Goldpin 2x1
JP3	Goldpin 2x1
JP1	Goldpin 5x2

Płytką drukowaną jest dostępna w Sklepie AVT jako AVT3276