

## Duży wyświetlacz 7-segmentowy RGB z diodami WS2812B



*Kupienie wyświetlaczy o dużych gabarytach nie jest problemem, gorzej gdy jest potrzebny wyświetlacz mogący świecić w różnych barwach. Można nabyć małe wyświetlacze z diodami czerwoną i zieloną, co pozwala uzyskać trzy podstawowe barwy oraz ich odcienie ale wyświetlaczy RGB raczej nie ma w sprzedaży. Nawet gdyby były do sterowania nimi potrzeba wielu przewodów. Prezentowany wyświetlacz z wbudowanym sterownikiem, można obsługiwać jednym wyprowadzeniem przez UART, dwoma przez I2C lub trzema za pośrednictwem SPI. Wyświetlacz pozwala na uzyskanie 24-bitowej palety barw.*

### Budowa i zasada działania

Schemat wyświetlacza pokazany jest na rysunku (plik „Protel Schematic v1.pdf”). Napięcie zasilające 5v doprowadzone jest do złącza J2. Dioda D1 zabezpiecza stabilizator U1 a co za tym idzie mikrokontroler U2 przed skutkami błędnego podłączenia zasilania. Napięcie 3,3V z wyjścia stabilizatora U1 zasila mikrokontroler, który steruje diodami wyświetlacza. W urządzeniu wykorzystano tani, dobrze wyposażony mikrokontroler ART SMT32G030F6P6, który przy cenie porównywalnej od Tiny3216 (także w obudowie 20 pin) oferuje:

	Tiny3216	SMT32G030F6P6
RAM	2kB	8kB
FLASH	32kB	32kB
Maksymalne taktowanie	20MHz	64MHz
Liczba poziomów przerwań	2	4
DMA	-	5 kanałów
UART	1	2
I2C	1 1MHz	2 1Mhz
SPI	1 10Mb/s	2 32Mb/s
Timery	4 trzy 16-bit jeden 12-bit	6 pięć 16-bit jeden systemowy w CPU 24-bit
WDG	1	2

\* Do porównania wybrano peryferia przydatne w projekcie sterownika wyświetlacza.

Jak można wywnioskować z tabelki, w takim projekcie należy zapomnieć o AVR ale i o Arduino! Jeśli będzie to AVR, to jest zbyt wolny, i nie ma DMA przez co jeszcze wolniejszy. Jeśli inne rozwiązania (np. ESP32), to zbyt drogie (koło 3 razy) w stosunku do STM32G030F6P6. Gdy ktoś się uprze, to i na AVR zrealizuje projekt, ale... w czasie transmisji do WS2812 zazwyczaj są zawieszanie przerwania. Jeśli transmisja jest realizowana na przerwaniach, co jest możliwe i było

pokazane w "Elektronice Dla Wszystkich 5-2020", to szybkość transferu po I2C, SPI, czy UART będzie ŻENUJĄCA ze względu na to, że komunikacja z WS2812 pochłonie ponad 90% czasu CPU, natomiast gdy na czas transmisji do WS2812 wyłączone będą przerwania, to ramki po I2C, SPI, UART będą gubione.

Jak widać, AVR są drogie jak na swoje możliwości. Xmega lepsze, ale drogie w stosunku do ARM i przy porównywalnej cenie mają mniejsze zasoby.

W wyświetlaczu 35 diod WS2812B jest ułożone na wzór wyświetlacza 7-segmentowego, dodatkowe trzy diody symbolizują kropkę. Diody są sterowane przez UART mikrokontrolera. Rezystor R2 podciąga wyjście UART do 5V co gwarantuje poprawne wysterowanie diody WS2812B. Jak to możliwe, że wyjście mikrokontrolera zasilanego z 3,3V można podciągnąć do 5V? Samo włączenie trybu otwarty dren (plik „gpio od.gif”) nie rozwiązałoby problemu, ponieważ na wyprowadzeniu UART znajdują się diody zabezpieczające (plik „basic io.gif”) przez co napięcie z rezystora R2 zostałoby ograniczone do napięcia zasilania powiększonego o spadek napięcia na diodzie zabezpieczającej. Prawdopodobnie to by wystarczyło, bo napięcie na takim wyprowadzeniu wynosiłoby 3,7...3,9V a diody WS2812 wymagają w stanie wysokim, napięcia wynoszącego 70% napięcia zasilania, co przy zasilaniu ich napięciem 5V daje 3,5V. Taka sytuacja miałaby miejsce w wielu starszych wersjach mikrokontrolerów w przypadku nowych, praktycznie wszystkie wyjścia akceptują napięcia wejściowe do 5,5V w związku z czym, diody zabezpieczające wejścia nie są podłączone do szyny napięcia zasilającego Vdd tylko do linii Vdd\_FT (rysunek „io\_g030.gif”). Dzięki takiemu rozwiązaniu, wyjścia w trybie otwarty dren, zostanie podciągnięte do 5V a nie 3,7...3,9V, gwarantując poprawną pracę diod WS2812B.

Złącze J3 umożliwia podłączenie wyświetlacza do mikrokontrolera. Kolejny wyświetlacz można podłączyć kaskadowo korzystając J4. Zworki JP1 ustalają adres wyświetlacza na szynie I2C w zakresie 0x20...0x2F. W przypadku wykorzystania do sterowania wyświetlaczem interfejsu UART, przy „adresowaniu” wyświetlacza, nie istotny jest jego adres, tylko miejsce wyświetlacza w łańcuchu, bowiem podobne jak w przypadku WS2812 tak w przypadku sterownika wyświetlacza, dane z wejścia są retransmitowane na wyjście po pominięciu danych przeznaczonych dla sterownika otrzymującego dane przez złącze J3.

Interfejs SPI podobnie jak UART retransmituje dane ale ze względu na jego specyfikę nie zaleca się łączenia wyświetlaczy w łańcuch tylko indywidualne uaktywnianie każdego mikrokontrolera. Dużą zaletą takiego rozwiązania jest możliwość podłączenia bardzo dużej liczby wyświetlaczy, które w przypadku I2C ograniczone są do 16, natomiast w przypadku UART ograniczeniem jest czas transmisji danych (dziewięć bajtów na wyświetlacz przy prędkości 115,2kb/s).

## Montaż i uruchomienie

Montaż jest typowy i nie wymaga szczegółowego omawiania. Urządzenie pobiera ok 9mA, w dużej mierze, za sprawą diody LED D2. Po wgraniu programu z wykorzystaniem złącza J1. Pobór prądu zwiększy się do 100-200mA bowiem, przy braku komunikacji, sterownik wyświetla na diodach DEMO (<https://youtu.be/fj-NMEIcshQ>). Maksymalny pobór prądu przez wyświetlacz (biały kolor, maksymalna jasność świecenia) wynosi około 2,3A.

Wyświetlacze najlepiej sterować przez interfejs I2C (maksymalnie 400kHz). W tym celu, wejście J3 pierwszego wyświetlacza podłączamy do mikrokontrolera natomiast wyjście J4 podłączamy z wejściem kolejnego wyświetlacza przewodem FLAT10 zakończonym z obu stron wtykami FC10. Należy jeszcze zadbać aby na każdym wyświetlaczu ustawić inny adres (adres bazowy, bez zworek to 0x20). Aby wysterować pierwszy wyświetlacz należy:

- Wygenerować warunek startu.
- Wysłać adres 0x20 do zapisu (bajt 0x40), jeśli nie ma ACK to błąd,
- Wysłać bajt reprezentujące zaświecone segmenty (bit0=A, bit1=B...bit6=G, bit7=kropka).
- Bajt koloru czerwonego,
- Bajt koloru zielonego,

- Bajt koloru niebieskiego.
- Warunek stopu.

Kolejne wyświetlacze obsługuje się tak samo tyle, że zmienia się adres urządzenia. Przykładowa transmisja danych do wyświetlacza przez I2C (plik „I2C.gif”).

Połączenia przy sterowaniu przez UART (**115200 8N1**) są podobne jak i przez I2C. Różnica jest w budowie ramki sterującej wyświetlaczami. Ramka dla jednego wyświetlacza składa się z:

Numer bajtu	Funkcja
0	Znacznik początku rekordu, Znak „:”
1	Starszy półbajt w kodzie HEXASCII zaświeconych segmentów
2	Młodszy półbajt w kodzie HEXASCII zaświeconych segmentów
3	Starszy półbajt w kodzie HEXASCII koloru czerwonego
4	Młodszy półbajt w kodzie HEXASCII koloru czerwonego
5	Starszy półbajt w kodzie HEXASCII koloru zielonego
6	Młodszy półbajt w kodzie HEXASCII koloru zielonego
7	Starszy półbajt w kodzie HEXASCII koloru niebieskiego
8	Młodszy półbajt w kodzie HEXASCII koloru niebieskiego
9	Znak CR o kodzie 13
10	Znak CR o kodzie 10

Znaki CR+LF opcjonalne LF+CR, oznaczają koniec ramki danych. Jeśli zabraknie tych znaków, ramka zostanie zinterpretowana po około 3ms od chwili nadania ostatniego znaku. Jeśli wysyłane są dane do kilku wyświetlaczy połączonych w łańcuch, CR+LF nadaje się po całej ramce danych. Przykładowa ramka dla trzech wyświetlaczy:

**:FF7f0000:01003f00:3f0000ff[CR+LF]**

spowoduje zaświecenie wszystkich segmentów z kropką (:FF) pierwszego wyświetlacza w kolorze czerwonym z połową mocy diody (red=7f, green=00, blue=00), segmentu „A” (:01) drugiego wyświetlacza w kolorze zielonym z 1/4 mocy diody (red=00, green=3f, blue=00) oraz cyfry zero (:3f – segmenty A...F) na trzecim wyświetlaczu w kolorze niebieskim z pełną mocą (red=00, green=00, blue=ff).

Sterując wyświetlaczem przez SPI, można je połączyć szeregowo używając takiej samej taśmy jak dla I2C czy UART ale takie sterowanie ma wady. Wynikają one z tego, że wyświetlacz na swoim wyjściu MISO wystawia dane z poprzedniej transmisji, oczywiście po pominięciu pierwszych czterech bajtów. Dlatego aby wysłać dane do kilku wyświetlaczy należy wysłać takie same ramki w liczbie równej wyświetlaczom. Dla ośmiu wyświetlaczy należy zatem wysłać osiem ramek.

Interfejs SPI pracuje w trybie **Mode0 z maksymalną prędkością 125kb/s**. Budowa ramki jest następująca:

Numer bajtu	Funkcja
0	Bajt reprezentujące zaświecone segmenty (bit0–A, bit1-B...bit6-G, bit7-kropka)
1	Intensywność koloru czerwonego
2	Intensywność koloru zielonego
3	Intensywność koloru niebieskiego

Dla kolejnych wyświetlaczy ramki należy powtórzyć. Cała ramka zaczyna się od poziomu niskiego

na linii wyboru „/SS” a kończy poziomem wysokim tej linii.

Lepszym rozwiązaniem niż łączenie sterowników w łańcuch jest indywidualne sterowanie każdym z osobna. Do tego nie można użyć prostej taśmy FLAT ponieważ do wszystkich sterowników należy doprowadzić linie SCK (to zapewnia taśma 1:1) oraz linie MOSI. Linii MISO nie używamy natomiast linie wyboru „/SS” należy przyłączyć do dekodera adresowego. Przykładowa transmisja danych do jednego wyświetlacza (plik „SPI 1.gif”) oraz trzech (plik „SPI 3.gif”) przy czym pokazano jedną transmisję a w wypadku trzech wyświetlaczy, transmisję należy powtórzyć trzykrotnie.

### **Który interfejs jest najlepszy do podłączenia wyświetlaczy?**

Wszystko zależy od wymagań. W wyborze może pomóc poniższa tabelka:

Interfejs	Liczba linii	Czas transmisji		
		1 wyświetlacz	10 wyświetlaczy	30 wyświetlaczy
UART 115,2kb/s	1	780us	7,8ms	23,4ms
I2C 400kHz	2	110us	920us	2,8ms (wymagany multiplexer i2C)
SPI 125kb/s indywidualne /SS	2 + liczba wyświetlaczy	250us	2,5ms	7,68ms
SPI łańcuch	3	250us	25ms	234ms

W stosunku do UART, I2C wymaga dwóch linii ale mikrokontroler nie musi mieć stabilnego sygnału zegarowego (ważne w starszych AVR) podobnie jak w przypadku SPI. SPI wydaje się najgorszy bo po połączeniu w łańcuch czasy transmisji są duże natomiast przy indywidualnym wyborze układu potrzeba dużej liczby linii sterujących ale SPI może być zbawienne, gdy nie ma dostępnego innego interfejsu.

Sas  
[sas@elportal.pl](mailto:sas@elportal.pl)

## Zestawienie elementów:

### Rezystory 1206:

R1 R2 1k

### Kondensatory 1206:

C1...C43 100nF ceramiczny

C34 10uF ceramiczny

### Kondensatory THT:

C35 100uF/10V Elektrolityczny

### Półprzewodniki:

U1 SPX1117-3.3 SOT-223

U2 STM32G030F6P6

D1 SS16

D2 Dioda LED Niebieska 1206

D3...D40 WS2812B

### Pozostałe:

J1 T821-1-06-S1

J2 TB-5.0-PP-2P + TB-5.0-PIN24

J3 ZL231-10PG

J4 ZL231-10PG

JP1 Goldpin 4x2 2,54mm