

USBtester

Monitor parametrów zasilania USB

Są takie projekty, których powstanie jest wynikiem potrzeby chwili. Tak właśnie było w tym przypadku, gdzie potrzebą była konieczność monitorowania parametrów zasilania urządzenia podłączonego do portu USB, dla którego tenże port stanowił źródło napięcia zasilania. Oczywiście, w takim wypadku można posłużyć się dowolnym multimetrem, lecz nie byłoby to rozwiązanie wygodne i bezpieczne.

Rekomendacje: monitor przyda się do monitorowania zasilania zestawów ewaluacyjnych lub innych urządzeń zasilanych za pomocą USB.

Postanowiłem skonstruować niewielki monitor zasilania, który włącza się między monitorowany port USB a zasilane urządzenie, pokazujący wszystkie parametry źródła zasilania w jednym czasie. Tak oto powstał monitor parametrów zasilania USB, które jest tematem artykułu. Jego schemat ideowy pokazano na **rysunku 1**.

Jest to nieskomplikowany system mikroprocesorowy, jego sercem jest niewielki mikrokontroler ATtiny44 odpowiedzialny za programową implementację interfejsu I²C, przy użyciu którego mikrokontroler realizuje obsługę układu INA226 będącego specjalizowanym, bardzo dokładnym, 16-bitowym, różnicowym przetwornikiem A/C oraz obsługę niewielkiego, acz bardzo efektywnego wyświetlacza OLED o rozdzielczości 128×32 piksele stanowiącego element graficznego interfejsu użytkownika. Wspomniany przetwornik A/C mierzy spadek napięcia na rezystorze szeregowym R1 (10 mΩ), dzięki czemu jest możliwe wyznaczenie prądu pobieranego przez urządzenie USB dołączone do portu USB_DEVICE. Nie jest to jednak zwykły, zewnętrzny przetwornik A/C, jakich wiele na rynku, ale specjalizowany układ przeznaczony do pomiaru prądu, napięcia i mocy urządzeń zasilanych napięciem stałym. Jako że jest to element wyjątkowy, warto choćby skrótowo zaznajomić się z jego specyfikacją.

Układ INA226 jest produkowany przez firmę Texas Instruments z przeznaczeniem do zastosowania w układach pomiaru prądu i mocy z wykorzystaniem

bocznika rezystancyjnego. Układ ten wyróżnia się następującymi, wybranymi cechami użytkowymi:

- Szeroki zakres napięć zasilania: 2,7...5,5 V.
- Duża dokładność pomiaru rzędu 0,1%.
- Możliwość pracy w systemach o szerokim zakresie napięcia szyny zasilającej 0...36 V.
- Możliwość pracy w konfiguracji low-side lub high-side.
- Bezpośredni pomiar napięcia, prądu i mocy.
- Konfigurowalny czas przetwarzania wbudowanego przetwornika A/C.
- Konfigurowalna funkcja uśredniania pomiarów.
- Dwa tryby pracy wbudowanego przetwornika A/C: ciągły i wyzwalany.
- Możliwość alarmowania po przekroczeniu zadanego poziomu prądu, napięcia szyny zasilającej odbiornik lub mocy pobieranej przez odbiornik.

Jak widać, układ INA226 idealnie wpisuje się w wymagania naszej aplikacji, oferując niespotykaną funkcjonalność i dokładność pomiarów. Schemat blokowy układu INA226 pokazano na **rysunku 2**.

Układ INA226 dokonuje ciągłego lub wyzwalanego przez aplikację pomiaru dwóch napięć: napięcia szyny zasilającej odbiornik (V_{BUS}) oraz napięcia na zaciskach bocznika rezystancyjnego (V_{SHUNT}), połączonych szeregowo z odbiornikiem. Na podstawie tych

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5672

Podstawowe parametry:

- Zakres pomiarowy napięcia: 0...10 V.
- Zakres pomiarowy prądu: 0...2 A.
- Zakres pomiarowy mocy: 0...10 W.
- Zakres pomiarowy energii: 0...10 Ah.
- Bazuje na układzie INA226.
- Komunikacja za pomocą mikrokontrolera ATtiny44.

Projekty pokrewne na www.media.avt.pl:

- Tester diod LED małej mocy (EP 2/2018)
- AVT-5488 Kontroler obciążenia portu USB (EP 2/2015)
- AVT-1823 Monitor prądu USB (EP 8/2014)
- AVT-5003 Tester elementów elektronicznych (EP 3/2001)

Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania!

Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
- wersja [A] – płytkę drukowaną bez elementów i dokumentacji kitu w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
 - wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja
 - wersja [UK] – zaprogramowany układ

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.

dwóch wielkości i zawartości rejestru konfiguracyjnego CALIBRATION, którego wartość zależy od wymaganej rozdzielczości pomiaru i parametrów zastosowanego bocznika

REKLAMA

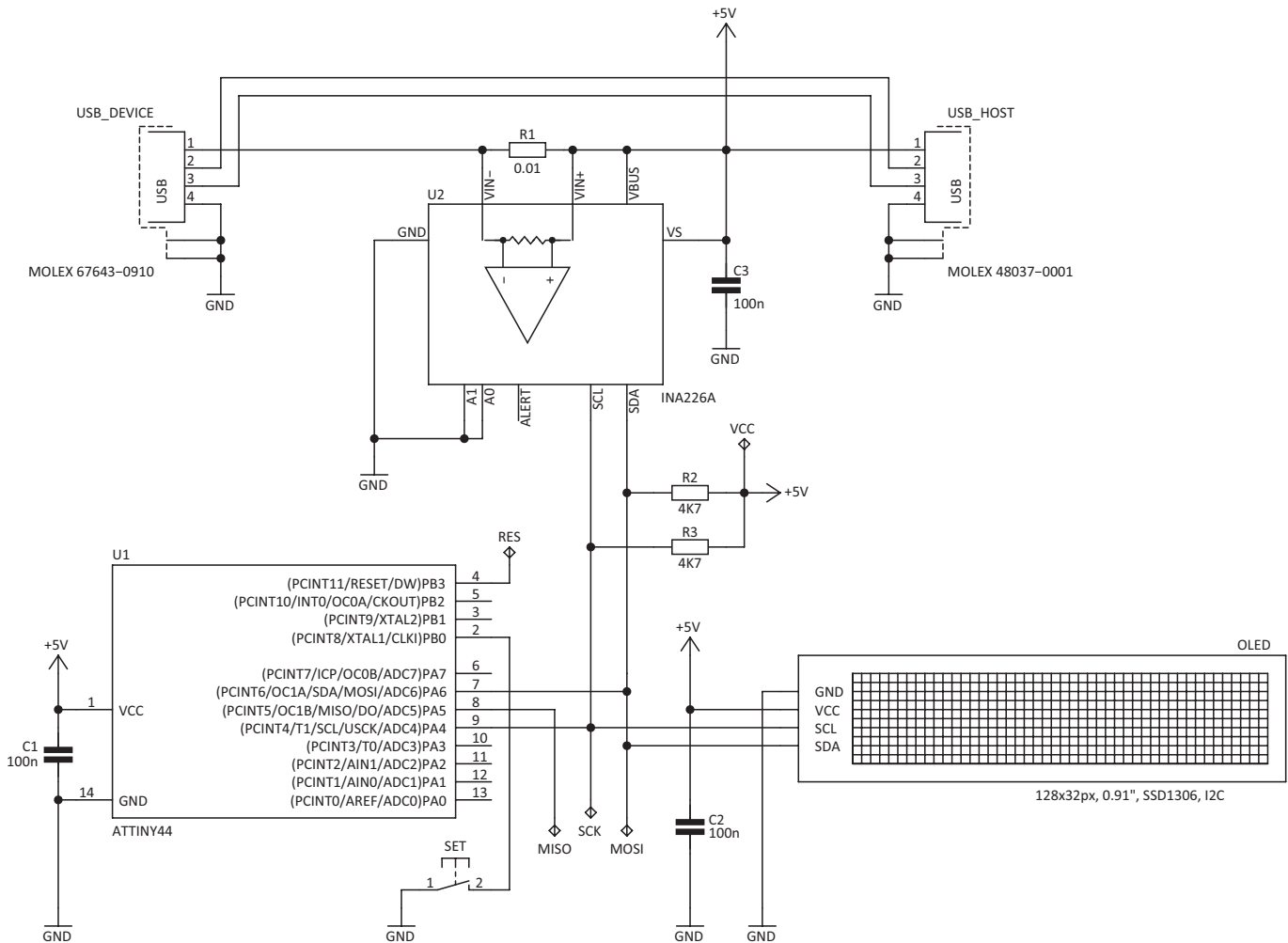
Specjalistyczne szkolenia dla elektroników i automatyków



TECHDAYS

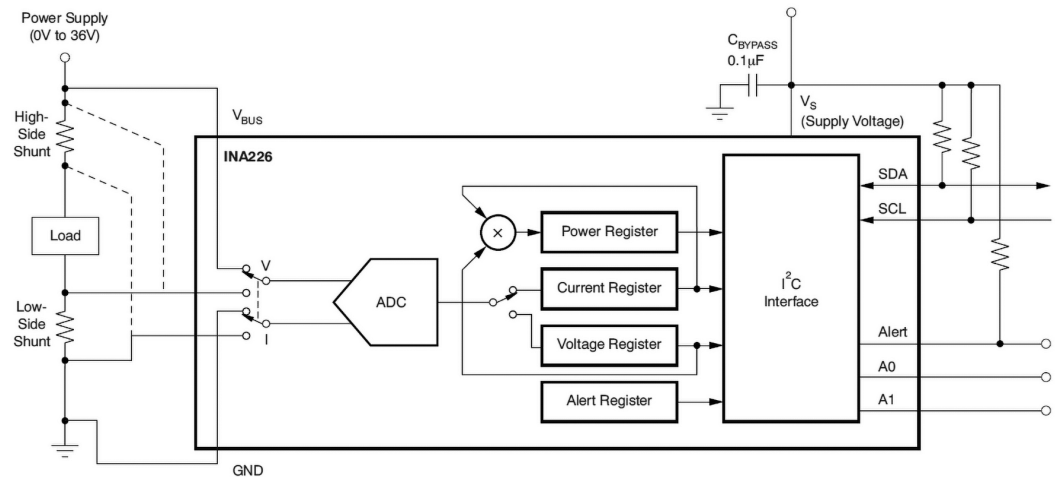
techdays@techdays.pl
TECHDAYS.PL

ST CERTYFIKOWANY PARTNER SZKOLENIOWY
life.augmented



Rysunek 1. Schemat ideowy urządzenia USBtester

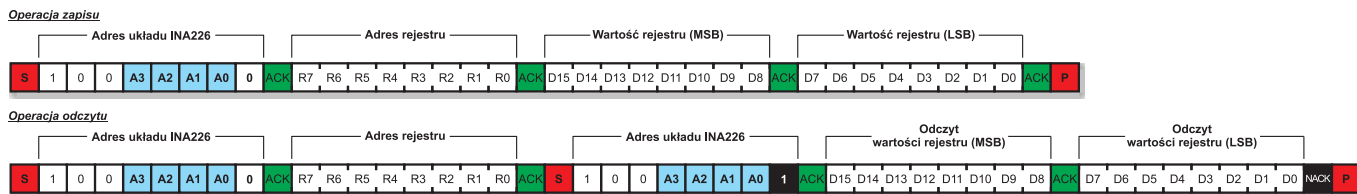
rezystancyjnego, układ oblicza prąd oraz moc pobieraną przez odbiornik i udostępnia je aplikacji użytkownika, ładując obliczone wielkości do stosownych rejestrów konfiguracyjnych, jak również ustawiając flagi zakończenia konwersji. Ponadto, dzięki wyposażeniu go w grupę specjalnych rejestrów konfiguracyjnych odpowiedzialnych za porównywanie zmierzonych i obliczonych wartości z wartościami progowymi, jak również wyjście ALERT, umożliwia generowanie



Rysunek 2. Schemat blokowy układu INA226

alarmów po przekroczeniu zdefiniowanych przez użytkownika progów: napięcia szyny zasilającej, napięcia na boczniku pomiarowym i mocy pobieranej przez odbiornik. Układ ma też możliwość niezależnej konfiguracji czasu przetwarzania przetwornika A/C, oddzielnie dla napięcia szyny zasilającej i napięcia bocznika rezystancyjnego oraz możliwość uśredniania pomiarów tych wielkości. Dzięki takiemu podejściu wydatnie zwiększono funkcjonalność użytkową układu oraz możliwość dostosowaniu trybu

Tabela 1. Znaczenie ustawień bitów M2...M0 rejestru CONFIGURATION	
M2...M0	Tryb pracy przetwornika A/C
000	Power-down
001	Pomiar napięcia bocznika (wyzwalany)
010	Pomiar napięcia szyny zasilającej (wyzwalany)
011	Pomiar napięcia bocznika i szyny zasilającej (wyzwalany)
100	Power-down
101	Pomiar napięcia bocznika (ciągły)
110	Pomiar napięcia szyny zasilającej (ciągły)
111	Pomiar napięcia bocznika i szyny zasilającej (ciągły)



S – sygnał Start magistrali I²C. **P** – sygnał Stop magistrali I²C. **ACK** – potwierdzenie po stronie odbiornika. **NACK** – brak potwierdzenia po stronie mikrokontrolera (koniec operacji odczytu). **A3, A2, A1, A0** – sprzętowy adres układu.

Rysunek 3. Budowa ramki danych magistrali I²C dla operacji zapisu i odczytu układu INA226

jego pracy do wymagań konkretnej aplikacji. Wydłużenie czasu przetwarzania zwiększa uzyskaną dokładność pomiaru, zaś uśrednianie większej liczby próbek zdecydowanie poprawia odstęp sygnału od szumu, w związku z czym w rzeczywistych aplikacjach należy dobierać maksymalne i możliwe do zaakceptowania wartości tych parametrów, kierując się, dla przykładu, szybkością zmian badanych przebiegów.

Aby poznać możliwości drzemiące w układzie INA226, warto przyjrzeć się rejestrom konfiguracyjnym. Będziemy korzystali z operacji zapisu i odczytu rejestrów konfiguracyjnych, więc na **rysunku 3** pokazano niezbędną sekwencję sygnałów magistrali I²C dla obu rodzajów operacji.

Rejestr: **CONFIGURATION [0x00]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
RST	X	X	X	AVG2	AVG1	AVG0	BUSCT2	BUSCT1	BUSCT0	SHCT2	SHCT1	SHCT0	M2	M1	M0

Rejestr **CONFIGURATION** służy do ustawienia podstawowych parametrów pracy wbudowanego w układ INA226 przetwornika

A/C. Znaczenie poszczególnych bitów jest następujące:

- **RST** – ustawienie tego bitu wymusza zerowanie układu INA226 i przywrócenie domyślnych wartości rejestrów.
- **AVG2...AVG0** – kombinacja bitów determinuje liczbę pomiarów wykonywanych przez przetwornik A/C w celu uśrednienia wartości wynikowych (napięć szyny zasilającej i bocznika) umieszczonych następnie w stosownych rejestrach danych. Uśrednianie coraz większej liczby pomiarów zwiększa wydatnie dokładność pomiarów, lecz z drugiej strony wydłuża sumaryczny czas konwersji, co nie zawsze jest dopuszczalne. Znaczenie poszczególnych

ustawień tychże bitów przedstawia się następująco (wynikiem ustawień jest liczba uśrednianych próbek):

000 → 1, 001 → 4, 010 → 16, 011 → 64, 100 → 128, 101 → 256, 110 → 512, 111 → 1024.

- **BUSCT2... BUSCT0** – ustawienia bitów determinują czas przetwarzania (konwersji) przetwornika A/C dla pomiaru napięcia szyny zasilającej V_{BUS} . Znaczenie poszczególnych ustawień bitów przedstawia się następująco: 000 → 140 μs, 001 → 204 μs, 010 → 332 μs, 011 → 588 μs, 100 → 1,1 ms, 101 → 2,116 ms, 110 → 4,156 ms, 111 → 8,244 ms.
- **SHCT2... SHCT0** – ustawienia bitów determinują czas przetwarzania (konwersji) przetwornika A/C dla pomiaru napięcia bocznika pomiarowego V_{SHUNT} . Znaczenie poszczególnych ustawień tychże bitów jest analogiczna do przypadku pomiaru napięcia szyny zasilającej.
- **M2... M0** – ustawienia bitów determinują tryb pracy przetwornika A/C.

REKLAMA

E-prenumerata to:

- najszybszy dostęp do nowego wydania magazynu
- wygodne archiwum na www.avt.pl
- hipertekstowy spis treści i wyszukiwarka
- wbudowane linki – klikasz i jesteś na odpowiedniej stronie WWW

Chcesz otrzymywać dodatek
Niezbędnik Elektronika?
Zamów prenumeratę
drukowaną na
www.avt.pl/prenumerata

Zamów e-prenumeratę (.pdf) na
www.avt.pl/prenumerata/elektroniczne

e-prenumerata roczna z rabatem

15% – 91,80 zł

e-prenumerata dwuletnia z rabatem

30% – 151,20 zł



Prenumeratory wersji drukowanej za równoległe e-wydania płać tylko 20% ceny: 21,60 zł/rok i 43,20 zł/2 lata

Znaczenie poszczególnych ustawień umieszczono w tabeli 1.

Rejestr: **SHUNT VOLTAGE [0x01]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
S	SD14	SD13	SD12	SD11	SD10	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0

Rejestr **SHUNT VOLTAGE** przechowuje 16-bitowy (ze znakiem, w standardzie U2) wynik przetwarzania napięcia bocznika pomiarowego V_{SHUNT} . Dostęp do tego rejestru z poziomu magistrali I²C nie podlega żadnym ograniczeniom, jednak wartość tam umieszczona jest prawidłowa dopiero po zakończeniu pomiaru stosownego napięcia (zakończeniu przetwarzania) lub po zakończeniu serii pomiarów, w przypadku wykorzystania mechanizmu uśredniania pomiarów. Wartość najmniej znaczącego bitu dla tego rejestru wynosi 2,5 μ V.

Rejestr: **BUS VOLTAGE [0x02]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	BD14	BD13	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0

Rejestr **BUS VOLTAGE** przechowuje 15-bitowy wynik przetwarzania napięcia szyny zasilającej V_{BUS} . Dostęp do tego rejestru z poziomu magistrali I²C nie podlega żadnym ograniczeniom, jednak wartość tam umieszczona jest prawidłowa dopiero po zakończeniu pomiaru stosownego napięcia (zakończeniu przetwarzania) lub po zakończeniu serii pomiarów, w przypadku wykorzystania mechanizmu uśredniania pomiarów. Wartość najmniej znaczącego bitu dla tego rejestru wynosi 1,25 mV.

Rejestr: **POWER [0x03]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

Rejestr **POWER** przechowuje 16-bitowy wynik obliczeń mocy pobieranej przez odbiornik. Dostęp do tego rejestru z poziomu magistrali I²C nie podlega żadnym ograniczeniom, jednak wartość tam umieszczona jest prawidłowa dopiero po zakończeniu pomiarów napięć szyny zasilającej i bocznika. Wartość najmniej znaczącego bitu dla tego rejestru zależy od wybranej przez użytkownika rozdzielczości pomiaru prądu odbiornika (CURRENT_LSB), o czym później, i wynosi $25 * \text{CURRENT_LSB}$.

Rejestr: **CURRENT [0x04]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
S	CD14	CD13	CD12	CD11	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0

Rejestr **CURRENT** przechowuje 16-bitowy (ze znakiem, w standardzie U2) wynik obliczeń prądu pobieranego przez odbiornik. Prąd ten obliczany jest na podstawie poniższej zależności (użyte nazwy reprezentują wartości umieszczone w stosownych rejestrach układu):

$$\text{CURRENT} =$$

$$(\text{SHUNT_VOLTAGE} * \text{CALIBRATION}) / 2048$$

Wartość najmniej znaczącego bitu tego rejestru równa jest wybranej w procedurze obliczania wartości dla rejestru kalibracyjnego rozdzielczości pomiaru prądu odbiornika (CURRENT_LSB). Dostęp do tego rejestru z poziomu magistrali I²C nie podlega żadnym

ograniczeniom, jednak wartość tam umieszczona jest prawidłowa dopiero po zakończeniu pomiaru napięcia bocznika.

Rejestr: **CALIBRATION [0x05]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	BD14	BD13	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0

Rejestr **CALIBRATION** przechowuje 15-bitową wartość kalibracji niezbędną do przeprowadzenia przez układ INA226 obliczeń prądu i mocy pobieranej przez odbiornik. Potrzeba wprowadzenia takiego, dodatkowego rejestru wynika z faktu, iż układ INA226 mierzy tak naprawdę wyłącznie napięcie bocznika i szyny zasilającej i na ich podstawie wyznacza inne wartości, w związku z czym, jeśli nie dysponowałby jakąkolwiek informacją o rezystancji zastosowanego

bocznika (i wymaganej rozdzielczości obliczeń), nie byłby w stanie dostarczyć gotowych wartości prądu i mocy pobieranej przez odbiornik. Aby temu sprostać, do rejestru CALIBRATION należy wpisać wartość, która zależna jest od oczekiwanej rozdzielczości pomiaru (czyli wartości najmniej znaczącego bitu) i wartości zastosowanego bocznika rezystancyjnego. Wartość rejestru CALIBRATION wyznaczamy na podstawie poniższego wzoru:

$$\text{CALIBRATION} = (5120 / (\text{CURRENT_LSB} * \text{SHUNT_RESISTOR}))$$

gdzie wartość CURRENT_LSB wyrażona jest w [mA], zaś wartość SHUNT_RESISTOR w [m Ω].

Rejestr: **MASK/ENABLE [0x06]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
SOL	SUL	BOL	BUL	POL	CNVR	X	X	X	X	X	AFF	CVRF	OVF	APOL	LEN

Rejestr **MASK/ENABLE** służy do konfiguracji bardzo ciekawej funkcjonalności, jaką udostępnia układ INA226, a mianowicie możliwości zgłaszania sprzętowego alarmu (wyjście ALERT) po przekroczeniu przez wybrane wielkości mierzone czy też obliczone zadanego progu wyzwania. Znaczenie

poszczególnych bitów tego rejestru przedstawia się następująco:

- **SOL** – ustawienie tego bitu powoduje wywołanie alarmu po przekroczeniu przez napięcie bocznika wartości progowej zdefiniowanej w rejestrze ALERT LIMIT.
- **SUL** – ustawienie tego bitu powoduje wywołanie alarmu w przypadku, gdy napięcie bocznika spadnie poniżej wartości progowej zdefiniowanej w rejestrze ALERT LIMIT.
- **BOL** – ustawienie tego bitu powoduje wywołanie alarmu po przekroczeniu

przez napięcie szyny zasilającej wartości progowej zdefiniowanej w rejestrze ALERT LIMIT.

- **BUL** – ustawienie tego bitu powoduje wywołanie alarmu w przypadku, gdy napięcie szyny zasilającej spadnie poniżej wartości progowej zdefiniowanej w rejestrze ALERT LIMIT.
- **POL** – ustawienie tego bitu powoduje wywołanie alarmu po przekroczeniu przez moc pobieraną przez odbiornik wartości progowej zdefiniowanej w rejestrze ALERT LIMIT.
- **CNVR** – ustawienie tego bitu powoduje wywołanie alarmu po zakończeniu bieżącej konwersji napięć (i wykonaniu stosownych obliczeń) i jest jednoznaczne z gotowością układu INA226 do wykonania kolejnych pomiarów.
- **AFF** – flaga, która jest ustawiana za każdym razem, gdy wystąpi alarm od przekroczenia wartości progowych. Pozwala na rozróżnienie przez aplikację użytkownika, czy wywołanie ALERT zgłosiło alarm od przekroczenia wartości progowych, czy po zakończeniu konwersji pomiarów napięć, w przypadku wykorzystywania tej ostatniej funkcjonalności.
- **CVRF** – flaga, która jest ustawiana po zakończeniu bieżącej konwersji napięć (to znaczy wszystkich konwersji w ramach całego procesu uśredniania pomiarów, jeśli korzysta się z tej funkcjonalności).
- **OVF** – flaga, która jest ustawiana w wypadku wystąpienia błędów obliczeń arytmetycznych. W takim wypadku wartości dostępne w rejestrach CURRENT I POWER mogą nie być poprawne.
- **APOL** – ustawienia tego bitu determinują polaryzację wyprowadzenia ALERT, czyli stan aktywny

Wykaz elementów:

Rezystory:

R1: 10 m Ω /500 mW/1% (pomiarowy w obudowie SMD1206)
R2, R3: 4,7 k Ω (SMD 0805)

Kondensatory:

C1...C3: 100 nF (SMD 0805)

Półprzewodniki:

U1: ATTiny44 (SOIC14)
U2: INA226A (MSOP-10)

Inne:

OLED: wyświetlacz OLED 128x32 piksele, 0,91", sterownik SSD1306, I²C, wymiary 38 mmx12 mm
SET: mikroprzełącznik SMD typu Omron B3U-3000PPM
USB_DEVICE: gniazdo żeńskie USB-A typu MOLEX 67643-0910
USB_HOST: gniazdo męskie USB-A typu MOLEX 48037-0001

w przypadku wystąpienia alarmu według następującej specyfikacji: 0 → aktywny stan niski, 1 → aktywny stan wysoki.

- **LEN** – ustawienia tego bitu determinują funkcjonalność zatrząskiwania stanu wyprowadzenia ALERT (i stosownych flag) po wystąpieniu alarmu. W razie wyzerowania tego bitu wyprowadzenie ALERT (i stosowne flagi) powróci do swojego stanu nieaktywnego, gdy tylko zdarzenie wywołujące alarm ustąpi. W przeciwnym wypadku wyprowadzenie ALERT i stosowne flagi pozostaną w stanie aktywnym do czasu wykonania operacji odczytu rejestru MASK/ENABLE.

Rejestr: **ALERT LIMIT [0x07]**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
AL15	AL14	AL13	AL12	AL11	AL10	AL9	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1	AL0

Rejestr **ALERT LIMIT** przechowuje 16-bitową wartość progu zadziałania dla mechanizmu alarmowania.

Uff, przebrnęliśmy przez lekturę dotyczącą właściwości układu INA226, ale moim

zdaniem było warto! Układ jest naprawdę arcyciekawy i dodatkowo dysponuje doskonałymi parametrami elektrycznymi. Pora więc na trochę praktyki w postaci źródeł programu obsługi tego peryferium. Na **listingu 1** przedstawiono zawartość pliku nagłówkowego modułu odpowiedzialnego za obsługę układu INA226, zaś na **listingu 2** funkcje pozwalających na odczyt i zapis rejestrów konfiguracyjnych układu.

Dalej, na **listingu 3** zamieszczono funkcje konfigurujące układ INA226, zaś na **listingach 4..6** odpowiedzialne za pomiar prądu, napięcia i mocy urządzenia monitorowanego.

W tym miejscu posiadamy już wszystkie, niezbędne informacje dotyczące zasady działania układu INA226, pora więc na przedstawienie szczegółów implementacyjnych dotyczących modułu zastosowanego wyświetlacza OLED, jako że on również stanowi bardzo ciekawe rozwiązanie układowe. Moduł ten wykorzystuje panel OLED o przekątnej 0,91" i rozdzielczości 128×32 piksele, wyposażony w popularny sterownik ekranu

SSD1306 skonfigurowany do pracy jako element sterowany za pomocą magistrali I²C. Jest to dość wygodne rozwiązanie, ponieważ wymaga zaangażowania tylko 2 wyprowadzeń mikrokontrolera do przeprowadzenia transmisji z modułem wyświetlacza, lecz niepozbawione wad, gdzie jedną z nich jest niewielka prędkość transmisji wpływająca na częstość odświeżania ekranu. Tradycyjnie zacznę od pokazania zawartości pliku nagłówkowego związanego z obsługą naszego peryferium, który to plik pokazano na **listingu 7**. Dalej, na **listingach 8 i 9** przedstawię podstawowe funkcje obsługi odpowiedzialne za przesłanie do sterownika SSD1306 rozkazu lub danej pamięci obrazu.

Jak widać, o tym, czy przesyłana dana jest wartością rejestru konfiguracyjnego, czy też zwyczajnie daną obrazu do wyświetlenia, decyduje wartość drugiego z przesyłanych bajtów danych (tuż po adresie urządzenia) nazywanego tutaj „Control byte”. Warto również podkreślić, że sterownik SSD1306 może przyjmować wiele następujących po sobie danych bez zmiany ich charakteru, tzn. w znaczeniu czy są wartościami kolejnych

Listing 1. Zawartość pliku nagłówkowego modułu obsługi układu INA226

```
#define SHUNT_RESISTOR_VALUE 10 // [mohm]
#define CHOSEN_CURRENT_LSB 1 // [mA]
#define CALCULATED_CALIBRATION_VALUE (5120/(CHOSEN_CURRENT_LSB* SHUNT_RESISTOR_VALUE))
#define INA226_WRITE_ADDR 0x80 // A1..A0 connected to GND
#define INA226_READ_ADDR 0x81 // A1..A0 connected to GND

#define INA226_CONFIG_REG 0x00
#define SOFTWARE_RESET (1<<15)
#define AVERAGES_1 (0<<9) //Default
#define AVERAGES_4 (1<<9)
#define AVERAGES_16 (2<<9)
#define AVERAGES_64 (3<<9)
#define AVERAGES_128 (4<<9)
#define AVERAGES_256 (5<<9)
#define AVERAGES_512 (6<<9)
#define AVERAGES_1024 (7<<9)
#define BUS_CONV_TIME_140US (0<<6)
#define BUS_CONV_TIME_204US (1<<6)
#define BUS_CONV_TIME_332US (2<<6)
#define BUS_CONV_TIME_588US (3<<6)
#define BUS_CONV_TIME_1100US (4<<6) //Default
#define BUS_CONV_TIME_2116US (5<<6)
#define BUS_CONV_TIME_4156US (6<<6)
#define BUS_CONV_TIME_8244US (7<<6)
#define SHUNT_CONV_TIME_140US (0<<3)
#define SHUNT_CONV_TIME_204US (1<<3)
#define SHUNT_CONV_TIME_332US (2<<3)
#define SHUNT_CONV_TIME_588US (3<<3)
#define SHUNT_CONV_TIME_1100US (4<<3) //Default
#define SHUNT_CONV_TIME_2116US (5<<3)
#define SHUNT_CONV_TIME_4156US (6<<3)
#define SHUNT_CONV_TIME_8244US (7<<3)
#define MODE_POWER_DOWN (0<<0)
#define MODE_SHUNT_TRIG (1<<0)
#define MODE_BUS_TRIG (2<<0)
#define MODE_SHUNT_BUS_TRIG (3<<0)
#define MODE_ADC_OFF (4<<0)
#define MODE_SHUNT_CONT (5<<0)
#define MODE_BUS_CONT (6<<0)
#define MODE_SHUNT_BUS_CONT (7<<0) //Default

#define INA226_SHUNT_VOLTAGE_REG 0x01
#define INA226_BUS_VOLTAGE_REG 0x02
#define INA226_POWER_REG 0x03
#define INA226_CURRENT_REG 0x04
#define INA226_CALIBRATION_REG 0x05

#define INA226_ALARM_ENABLE_REG 0x06
#define SHUNT_OVER_VOLTAGE_ALARM (1<<15)
#define SHUNT_UNDER_VOLTAGE_ALARM (1<<14)
#define BUS_OVER_VOLTAGE_ALARM (1<<13)
#define BUS_UNDER_VOLTAGE_ALARM (1<<12)
#define OVER_POWER_ALARM (1<<11)
#define CONVERSION_READY_ALARM (1<<10)
#define ALERT_FUNCTION_FLAG (1<<4)
#define CONVERSION_READY_FLAG (1<<3)
#define MATH_OVERFLOW_FLAG (1<<2)
#define ALARM_PIN_ACTIVE_HIGH (1<<1)
#define ALARM_PIN_ACTIVE_LOW (0<<1)
#define ALARM_LATCH_ENABLED (1<<0)
#define ALARM_LATCH_DISABLED (0<<0)

#define INA226_ALERT_LIMIT_REG 0x07
#define INA226_DIE_ID_REG 0xFF
```

Ustawienia Fuse bitów:

```
CKSEL3...0: 0010
SUT1...0: 10
CKDIV8: 0
CKOUT: 1
DWEN: 1
EESAVE: 0
```

Listing 2. Funkcje zapisujące i odczytujące rejestry konfiguracyjne układu INA226

```
void INA226writeRegister(uint8_t registerAddr, uint16_t registerValue)
{
    i2cStart();
    i2cWriteByte(INA226_WRITE_ADDR); //INA226 WR address
    i2cWriteByte(registerAddr); //Written register address
    i2cWriteByte(registerValue>>8); //Register MSB value
    i2cWriteByte(registerValue & 0xFF); //Register LSB value
    i2cStop();
}

uint16_t INA226readRegister(uint8_t registerAddr)
{
    register uint16_t readValue;

    i2cStart();
    i2cWriteByte(INA226_WRITE_ADDR); //INA226 WR address
    i2cWriteByte(registerAddr); //Written register address
    i2cRepeatStart(); //Restart
    i2cWriteByte(INA226_READ_ADDR); //INA226 RD address
    readValue = i2cReadByte(ACK)<<8;
    readValue |= i2cReadByte(NACK);
    i2cStop();
    return readValue;
}
```

Listing 3. Funkcja konfigurująca układ INA226

```
void INA226init(uint16_t Configuration)
{
    //Konfigurowanie INA226
    INA226writeRegister(INA226_CONFIG_REG, Configuration);
    //Kalibrowanie INA226
    INA226writeRegister(INA226_CALIBRATION_REG, CALCULATED_CALIBRATION_VALUE);
}
```

Listing 4. Funkcja pomiaru prądu

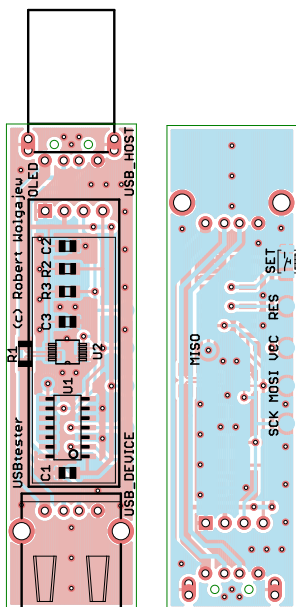
```
uint16_t INA226readCurrent(void) //CHOSEN_CURRENT_LSB = 1mA
{
    register uint16_t readValue;

    readValue = INA226readRegister(INA226_CURRENT_REG);
    //Dla wartości ujemnych przyjmujemy 0
    if(readValue & (1<<15)) readValue = 0;
    return readValue;
}
```

rejestrów konfiguracyjnych, czy też kolejnymi danymi obrazu. Krótko mówiąc, wysyłamy odpowiednią wartość bajta „Control byte” a po nim szereg kolejno następujących po sobie wartości pamięci ekranu. Tego rodzaju mechanizm, co nie jest bez znaczenia, wydatnie przyspiesza wyświetlanie danych. Pora na przedstawienie funkcji inicjalizacyjnej, którą pokazano na **listingu 10**. Dalej, na **listingu 11** zamieszczono funkcję narzędziową odpowiedzialną za ustawienie aktywnego obszaru ekranu, w ramach którego przeprowadzany jest zapis do pamięci ekranu sterownika SSD1306. Tego typu rozwiązanie znacznie przyspiesza proces wyświetlania danych obrazu, gdyż w pierwszej kolejności definiujemy aktywny obszar pamięci ekranu sterownika SSD1306, zaś w drugim zwyczajnie przesyłamy szereg następujących po sobie bajtów danych obrazu, które zostaną niejako „ułożone” w zdefiniowanym wcześniej obszarze, standardowo począwszy od lewej do prawej i z góry na dół. Kolejność umieszczania wspomnianych danych podlega konfiguracji, przez co możemy w łatwy sposób operować obracaniem wyświetlanych danych. Funkcję pozwalającą na wspomniane wcześniej operowanie obracaniem ekranu pokazano na **listingu 12**.

Czas na funkcje odpowiedzialne za rysowanie prostych elementów graficznych, to jest funkcję odpowiedzialną za wyświetlanie obrazków na ekranie wyświetlacza OLED oraz funkcję pozwalającą na rysowanie znaków, z użyciem bieżącej czcionki ekranowej, której parametry przechowuje globalna zmienna CurrentFont. Wspomniane funkcje pokazano na **listingu 13** i **14**. Do pokazania pozostaje funkcja ustawiająca parametry bieżącej czcionki ekranowej – **listing 15**.

To tyle, jeśli chodzi o obsługę tego niezmiernie ciekawego panelu OLED, który



Rysunek 4. Schemat montażowy urządzenia USBtester

```
Listing 5. Funkcja pomiaru napięcia zasilającego
uint16_t INA226readBusVoltage(void) // [mV]
{
    register uint16_t readValue;

    readValue = INA226readRegister(INA226_BUS_VOLTAGE_REG); //LSB = 1.25mV
    readValue += (readValue/4); //readValue = 1.25*readValue -> otrzymamy wartość w [mV]
    return readValue;
}
```

```
Listing 6. Funkcja pomiaru mocy
uint16_t INA226readPower(void) // [mW]
{
    register uint16_t readValue;

    readValue = INA226readRegister(INA226_POWER_REG)*25; //LSB = 25mW
    return readValue;
}
```

z uwagi na doskonałą jakość obrazu, bardzo niską cenę i niewielkie rozmiary może znaleźć zastosowanie w niejednym praktycznym projekcie. Przejdźmy zatem do tytułowego urządzenia, którego schemat montażowy pokazano na **rysunku 4**.

Jak widać, zaprojektowano bardzo zgrabną, niewielką płytkę drukowaną, która swoim kształtem przypomina typowe urządzenie podłączone do portu USB. Montaż układu rozpoczynamy od przylutowania elementu INA226. Proces ten najłatwiej wykonać przy użyciu stacji lutowniczej za gorące powietrze (tzw. Hot Air) i odpowiednich stopów lutowniczych. Jeśli jednak nie dysponujemy tego rodzaju sprzętem, można również zastosować metodę z wykorzystaniem typowej stacji lutowniczej. Najprostszym sposobem montażu elementów o tak dużym zagęszczeniu wyprowadzeń, niewymagającym jednocześnie posiadania specjalistycznego sprzętu, jest użycie typowej stacji lutowniczej, dobrej jakości cyny z odpowiednią ilością topnika oraz dość cienkiej plecionki rozlutowniczej,

```
Listing 7. Plik nagłówkowy modułu obsługi wyświetlacza OLED
//OLED I2C WRITE ADDRESS
#define OLED_WR_ADDRESS 0x78
//DATA/COMMAND BYTE DEFINITIONS
#define DATA_BYTE 0x40
#define COMMAND_BYTE 0x00
//OLED COMMANDS
#define SET_CONTRAST_CMD 0x81
#define DISPLAY_ALLON_RESUME_CMD 0xA4
#define DISPLAY_ALLON_CMD 0xA5
#define NORMAL_DISPLAY_CMD 0xA6
#define INVERSE_DISPLAY_CMD 0xA7
#define DISPLAY_OFF_CMD 0xAE
#define DISPLAY_ON_CMD 0xAF
#define SET_DISPLAY_OFFSET_CMD 0xD3
#define SET_COM_PINS_CMD 0xDA
#define SET_VCOMH_DESELECT_CMD 0xDB
#define SET_DISPLAY_CLOCK_DIV_CMD 0xD5
#define SET_PRECHARGE_PERIOD_CMD 0xD9
#define SET_MULTIPLEX_RATIO_CMD 0xA8
#define SET_START_LINE_CMD 0x40
#define MEMORY_MODE_CMD 0x20
#define SET_COLUMN_ADDR_CMD 0x21
#define SET_PAGE_ADDR_CMD 0x22
#define SET_COM_SCAN_NORMAL_CMD 0xC0
#define SET_COM_SCAN_REMAPMED_CMD 0xC8
#define SET_SEG_REMAPING_CMD 0xA0
#define CHARGEPUUMP_CMD 0x8D
```

```
Listing 8. Funkcja przesyłająca słowo konfiguracji do SSD1306
void SSD1306writeCmd(uint8_t Command)
{
    i2cStart();
    i2cWriteByte(OLED_WR_ADDRESS);
    i2cWriteByte(COMMAND_BYTE); //Control byte
    i2cWriteByte(Command); //Data byte
    i2cStop();
}
```

```
Listing 9. Funkcja przesyłająca dane do pamięci obrazu SSD1306
void SSD1306writeData(uint8_t Data)
{
    i2cStart();
    i2cWriteByte(OLED_WR_ADDRESS);
    i2cWriteByte(DATA_BYTE); //Control byte
    i2cWriteByte(Data); //Data byte
    i2cStop();
}
```

```
Listing 10. Funkcja inicjująca wyświetlacz OLED
void OLEDinit(void)
{
    SSD1306writeCmd(DISPLAY_OFF_CMD);
    SSD1306writeCmd(SET_START_LINE_CMD | 0x00);
    SSD1306writeCmd(SET_CONTRAST_CMD);
    SSD1306writeCmd(0x8F); //Contrast
    SSD1306writeCmd(SET_SEG_REMAPING_CMD | 0x00);
    SSD1306writeCmd(SET_COM_SCAN_NORMAL_CMD);
    SSD1306writeCmd(NORMAL_DISPLAY_CMD);
    SSD1306writeCmd(SET_MULTIPLEX_RATIO_CMD);
    SSD1306writeCmd(0x1F);
    SSD1306writeCmd(SET_DISPLAY_OFFSET_CMD);
    SSD1306writeCmd(0x00); //No offset
    SSD1306writeCmd(SET_DISPLAY_CLOCK_DIV_CMD);
    SSD1306writeCmd(0x80); //Set divide ratio, Set Clock as 100 Frames/Sec
    SSD1306writeCmd(SET_PRECHARGE_PERIOD_CMD);
    SSD1306writeCmd(0xF1); //Set Pre-Charge as 15 Clocks & Discharge as 1 Clock
    SSD1306writeCmd(SET_COM_PINS_CMD);
    SSD1306writeCmd(0x02);
    SSD1306writeCmd(SET_VCOMH_DESELECT_CMD); //---set vcomh
    SSD1306writeCmd(0x40); //Set VCOM Deselect Level
    SSD1306writeCmd(MEMORY_MODE_CMD);
    SSD1306writeCmd(0x00); //Horizontal addressing mode
    SSD1306writeCmd(CHARGEPUUMP_CMD);
    SSD1306writeCmd(0x14); //Disable
    SSD1306writeCmd(DISPLAY_ALLON_RESUME_CMD);
    SSD1306writeCmd(NORMAL_DISPLAY_CMD);
    SSD1306writeCmd(DISPLAY_ON_CMD);
}
```

Listing 11. Funkcja ustawiająca aktywny obszar ekranu

```
//Column: 0...127, Page: 0...3
void OLEDSetActiveWindow(uint8_t startColumn, uint8_t startPage, uint8_t endColumn, uint8_t endPage)
{
    SSD1306writeCmd(SET_COLUMN_ADDR_CMD);
    SSD1306writeCmd(startColumn);
    SSD1306writeCmd(endColumn);
    SSD1306writeCmd(SET_PAGE_ADDR_CMD);
    SSD1306writeCmd(startPage);
    SSD1306writeCmd(endPage);
}
```

Listing 12. Funkcja obracająca wyświetlany obraz

```
void OLEDsetRotation(uint8_t Rotation)
{
    if(Rotation)
    {
        SSD1306writeCmd(SET_SEG_REMAPING_CMD | 0x00);
        SSD1306writeCmd(SET_COM_SCAN_NORMAL_CMD);
    }
    else
    {
        SSD1306writeCmd(SET_SEG_REMAPING_CMD | 0x01);
        SSD1306writeCmd(SET_COM_SCAN_REMAPED_CMD);
    }
}
```

Listing 13. Funkcja wyświetlająca bitmapy

```
void OLEDdrawBitmap(uint8_t Column, uint8_t Page, const uint8_t *Bitmap)
{
    register uint8_t Width, Height;
    register uint16_t bytesToSend;

    Width = pgm_read_byte(Bitmap++); //Pierwszy bajt tablicy Bitmap to szerokość:
0...127
    Height = pgm_read_byte(Bitmap++)>>3; //Drugi bajt tablicy Bitmap to wysokość: 8, 16,
24...64 -> przeliczamy na bajty
    bytesToSend = Width*Height; //Liczba bajtów przeznaczonych do wysłania do OLEDa
    OLEDSetActiveWindow(Column, Page, Column+Width-1, Page+Height-1);
    i2cStart();
    i2cWriteByte(OLED_WR_ADDRESS);
    i2cWriteByte(DATA_BYTE); //Control byte
    while(bytesToSend--) i2cWriteByte(pgm_read_byte(Bitmap++));
    i2cStop();
}
```

Listing 14. Funkcja rysująca znaki z użyciem bieżącej czcionki

```
void OLEDdrawChar(char Character, uint8_t Column, uint8_t Page)
{
    const uint8_t *dataPointer;
    register uint8_t bytesToSend;

    //Ustalamy adres początku wzorca znaku ASCII, który zamierzamy wyświetlić
    dataPointer = &CurrentFont.Bitmap[(CurrentFont.BytesPerChar*(Character-CurrentFont.FirstCharCode))];
    //Określamy okno zapisu by uprościć samą procedurę zapisu
    OLEDSetActiveWindow(Column, Page, Column+CurrentFont.Width-1, Page+CurrentFont.Height-1);
    //Określamy liczbę bajtów do wysłania
    bytesToSend = CurrentFont.BytesPerChar;
    i2cStart();
    i2cWriteByte(OLED_WR_ADDRESS);
    i2cWriteByte(DATA_BYTE); //Control byte
    while(bytesToSend--) i2cWriteByte(pgm_read_byte(dataPointer++));
    i2cStop();
}
```

Listing 15. Funkcja ustawiająca parametry bieżącej czcionki ekranowej

```
void OLEDsetFont(const fontDescription *Font)
{
    //Rzeczywista szerokość czcionki
    CurrentFont.Width = pgm_read_byte(&Font->Width);
    //Rzeczywista wysokość czcionki
    CurrentFont.Height = pgm_read_byte(&Font->Height);
    //Odstęp pomiędzy znakami
    CurrentFont.Interspace = pgm_read_byte(&Font->Interspace);
    //Liczba bajtów na definicję pojedynczego znaku
    CurrentFont.BytesPerChar = pgm_read_byte(&Font->BytesPerChar);
    //Kod ASCII definicji pierwszego znaku
    CurrentFont.FirstCharCode = pgm_read_byte(&Font->FirstCharCode);
    //Wskaźnik do tablicy wzorców tej czcionki
    CurrentFont.Bitmap = (uint8_t*)pgm_read_word(&Font->Bitmap);
}
```

która umożliwi usunięcie nadmiaru cyny spomiędzy wyprowadzeń złącza. Należy przy tym uważać, by nie uszkodzić termicznie tego elementu. Następnie lutujemy pozostałe elementy półprzewodnikowe, potem rezystory i kondensatory oraz inne elementy bierne

które umożliwi usunięcie nadmiaru cyny spomiędzy wyprowadzeń złącza. Należy przy tym uważać, by nie uszkodzić termicznie tego elementu. Następnie lutujemy pozostałe elementy półprzewodnikowe, potem rezystory i kondensatory oraz inne elementy bierne

tualnych zwarc. Wspomniana kontrola będzie znacznie łatwiejsza, jeśli zmontowaną płytkę przemyjemy alkoholem izopropylowym w celu wypłukania nadmiaru kalafonii lutowniczej. Na samym końcu, do tak przygotowanej płytki, montujemy wyświetlacz OLED, zwyczajnie lutując jego wyprowadzenia w przeznaczone do tego celu pola lutownicze (należy sprawdzić polaryzację zasilania), gdyż połączenia te zapewniają mu jednocześnie wystarczający montaż mechaniczny. Poprawnie zmontowany układ powinien działać tuż po włączeniu zasilania. Na rysunku **rysunku 5** pokazano zmontowane urządzenie przed przylutowaniem wyświetlacza OLED.

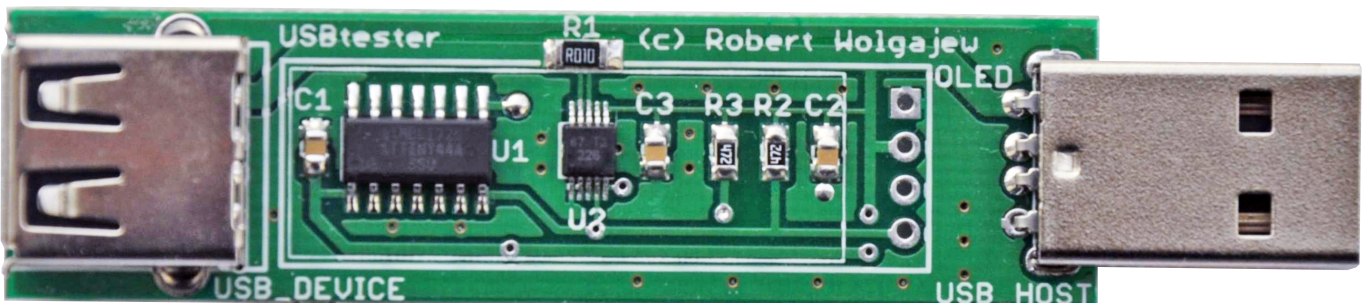
Wygląd graficznego interfejsu użytkownika pokazano na **rysunku 6**. Wartość napięcia źródła zasilania (w tym przypadku portu USB), prąd i moc pobierane przez podłączone urządzenie USB oraz energia elektryczna przekazana do urządzenia są pokazywane

na wyświetlaczu. Wszystkie wartości są odświeżane 5 razy na sekundę. Dodatkowo, w naszym urządzeniu zamontowano przycisk SET, który ma dwojakie zastosowanie. Jego krótkie przyciśnięcie powoduje każdorazowe obrócenie wyświetlanych danych o kąt 180°, co może być użyteczne w wypadku różnego położenia złącza USB, do którego dołączamy monitor. Długie przyciśnięcie przycisku powoduje wyzerowanie licznika energii przekazanej do odbiornika.

Robert Wołgajew, EP

5.200V | 1.040W
0.200A | 2.040Ah

Rysunek 6. Wygląd interfejsu użytkownika urządzenia USBtester



Rysunek 5. Zmontowane urządzenie USBtester tuż przed przylutowaniem wyświetlacza OLED