

Volta – miniaturowy woltomierz

Inspiracją do powstania niniejszego projektu był inny mój projekt o nazwie Ampera opublikowany na łamach „Elektroniki Praktycznej” w numerze 5/2020. Wspomniany układ jest amperomierzem prądu stałego mającym możliwość alarmowania po przekroczeniu konfigurowalnego progu. Wśród niewielkich przyrządów warsztatowych dobrze byłoby mieć również woltomierz napięcia stałego o podobnej konstrukcji i właśnie w ten sposób narodził się projekt nazwany Volta.

Systemem steruje niewielki, ale nowoczesny mikrokontroler firmy Microchip (dawniej Atmel) typu ATXmega8E5. Wybór tego konkretnego układu z szerokiej palety rodziny AVR nie był przypadkowy. Zależało mi na tym, aby urządzenie odznaczało się dużą dokładnością pomiaru nieokupioną stosowaniem zewnętrznych przetworników ADC. To spowodowało, że odrzuciłem starsze konstrukcje mikrokontrolerów AVR (ATmega i ATtiny), gdyż wyposażono je w niezbyt dokładny, jak na nasze potrzeby, 10-bitowy przetwornik ADC oraz nieprecyzyjne źródło napięcia odniesienia.

Uwagę skierowałem na najnowsze konstrukcje mikrokontrolerów AVR z rodziny ATXmega, serii E. W przypadku tych układów mamy do dyspozycji wbudowany, 12-bitowy przetwornik ADC o zdecydowanie lepszych parametrach elektrycznych, jak i dość dokładne (w porównaniu z poprzednim) źródło napięcia odniesienia, z którego w końcowej aplikacji jednak nie skorzystałem. Zastosowałem tutaj zewnętrzną, bardzo dokładną referencję pod postacią układu LM4040CYM3-2.5 (U3), który nie

dość, że bardzo tani, to dodatkowo odznacza się świetnymi parametrami elektrycznymi, co najmniej o rząd wielkości lepszymi niż źródła wbudowane w strukturę mikrokontrolera ATXmega.

Budowa i działanie

Schemat ideowy układu pokazano na rysunku 1. Jest to bardzo prosty system mikroprocesorowy, w którym wykorzystano niewielką część wyprowadzeń mikrokontrolera i jego zasobów sprzętowych. Jednak mikrokontroler realizuje następujące zadania: wykonuje cykliczny pomiar napięcia (10 razy/s) doprowadzonego z wejścia VIN poprzez rezystancyjny dzielnik napięcia na wejście PA7 układu. Pomiar wykonywany jest za pomocą 12-bitowego przetwornika ADC pracującego w trybie single-ended. Co ciekawe i charakterystyczne dla mikrokontrolerów typu ATXmega, cała akwizycja danych odbywa się w pełni automatycznie, przy użyciu zasobów sprzętowych układu. Jest to możliwe dzięki powiązaniu ze sobą kilku podsystemów mikrokontrolera: timera TCC5, kanału 0 systemu zdarzeń oraz przetwornika

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

W ofercie AVT* AVT-5820

Podstawowe parametry:

- zakres pomiarowy: 0...50 V,
- rozdzielczość pomiaru: 20 mV,
- dokładność pomiaru: 1%,
- zakres ustawień dla funkcji alarmowania: 0...50 V,
- rozdzielczość ustawień dla funkcji alarmowania: 10 mV,
- zasilanie: 4,5...10 V, ok. 25 mA.

Projekty pokrewne na www.media.avt.pl:

- AVT-5766 Ampera – miniaturowy miernik prądu stałego (EP 5/2020)
- Projekt 233 3-fazowy woltomierz TRMS (EP 10/2017)
- Projekt 232 Miłiamperomierz TRMS (EP 9/2017)
- AVT-5507 Miernik UIPTR (EP 7/2015)
- AVT-5399 Dwukanałowy multimetr panelowy (EP 6/2013)
- AVT-5386 Podwójny woltomierz i amperomierz (EP 3/2013)
- AVT-5383 Miernik tablicowy UIPT (EP 2/2013)
- AVT-5339 Woltomierz cyfrowy (EP 4/2012)
- AVT-5333 Multimetr panelowy (EP 3/2012)
- AVT-5318 Miernik mocy skutecznej wzmacniacza audio (EP 11/2011)
- AVT-5300 VMOD – Uniwersalny miernik napięcia (EP 7/2011)
- AVT-5233 3-kanałowy woltomierz (EP 5/2010)
- AVT-5182 Wielokanałowy rejestrator napięć (EP 4/2009)
- AVT-2857 Moduł woltomierza/amperomierza (Edw 3/2008)

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] – zamontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
• wersja [A] – płytkę drukowaną bez elementów i dokumentacji Kity 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 na 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.

ADC. Powiązanie pokazano schematycznie na **rysunku 2**.

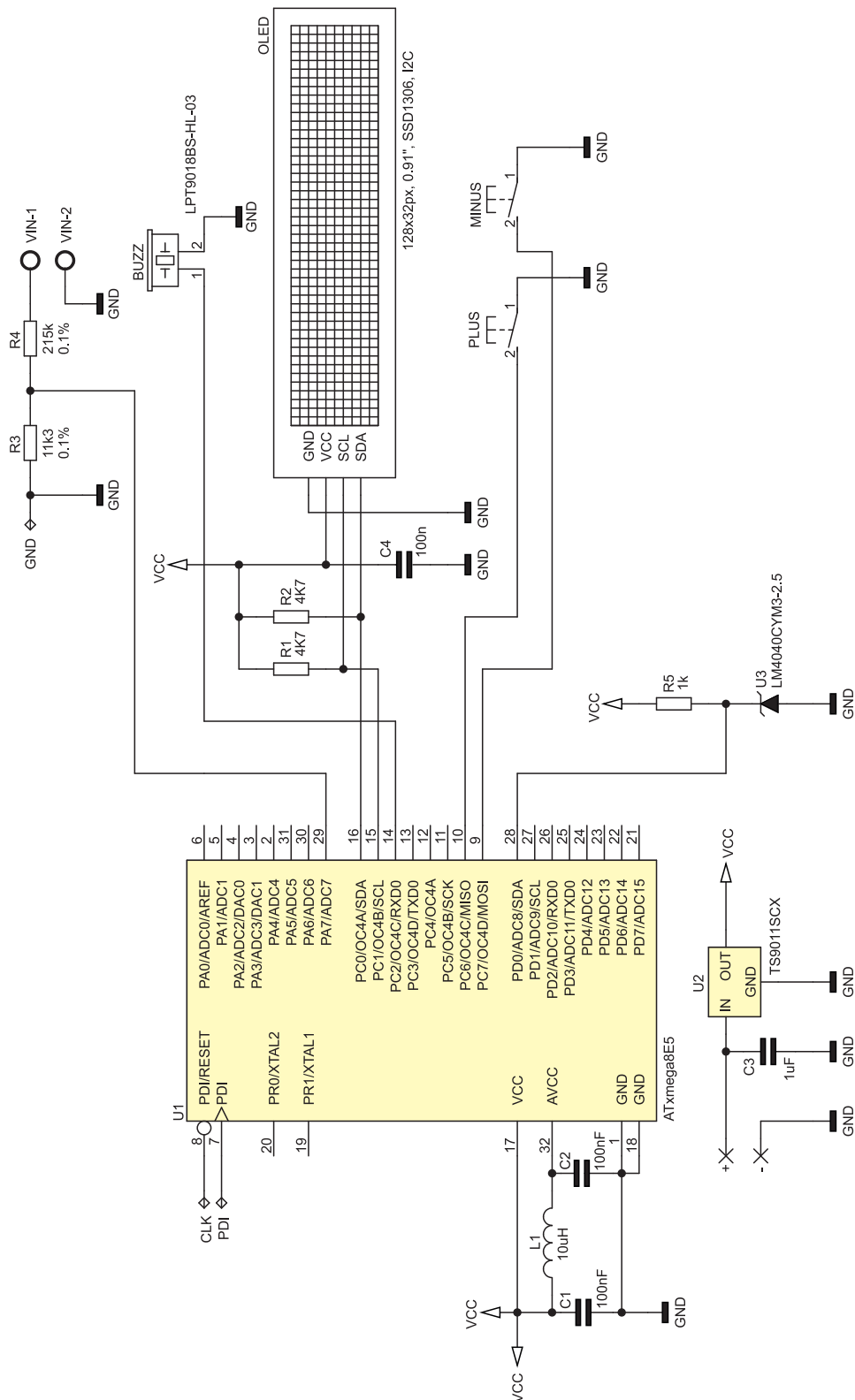
Podstawowym podsystemem mikrokontrolera ATXmega8E5 inicjującym pomiar przetwornika ADC jest układ czasowo-licznikowy TCC5, który jest taktowany wysokostabilnym przebiegiem zegarowym o częstotliwości 250 kHz. Sygnał jest uzyskiwany dzięki podzieleniu zegara 2 MHz taktującego rdzeń mikrokontrolera przez preskaler o podziale 8. Zdarzenie przepełnienia licznika (OVF) jest generowane co 25000 taktów zegara (PER=24999), czyli 10 razy na sekundę, i jest połączone z kanałem CH0 systemu zdarzeń (EVENT SYSTEM), dla którego stanowi źródło zdarzeń. Kanał CH0 systemu zdarzeń jest z kolei wyzwalaczem (TRIG) dla wbudowanego w strukturę mikrokontrolera przetwornika ADC, dzięki czemu możliwe jest próbkowanie wejściowego sygnału VIN w równych odstępach czasu (10 Hz). Dalej przerwanie od zakończenia konwersji przetwornika ADC (*ADCA_CH0_vect*) odpowiedzialne jest wyłączenie za ustawienie flagi programowej *ADCCready* dla programu obsługi aplikacji.

Warto również wspomnieć o innej możliwości automatycznej akwizycji danych, którą początkowo wykorzystywałem w programie obsługi aplikacji. Mowa o podsystemie EDMA, dzięki któremu możliwa staje się akwizycja danych i ich automatyczne umieszczanie w buforze programowym bez udziału rdzenia mikrokontrolera. Dzieje się tak dzięki specjalnym wyzwaczom podsystemu EDMA, którym dla przykładu może być zdarzenie zakończenia konwersji przetwornika ADC. Muszę przyznać, że początkowo wykorzystywałem tę zaawansowaną możliwość mikrokontrolera rodziny Xmega, lecz z uwagi na fakt, iż program obsługi aplikacji nie wykonuje absolutnie żadnych zadań poza zbieraniem i wyświetlaniem danych w końcowym programie, zrezygnowałem z usług podsystemu EDMA, gdyż nie wnosił on żadnych usprawnień. Ostatecznie system akwizycji urządzenia Volta wygląda dokładnie tak, jak to pokazano na rysunku 2.

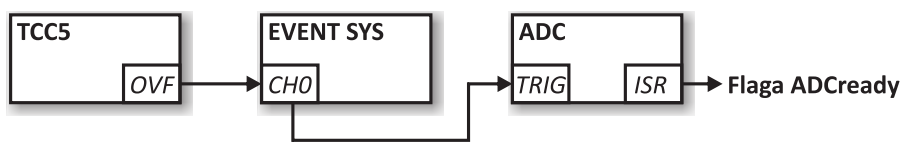
Kod odpowiedzialny za inicjalizację układu czasowo-licznikowego TCC5 pokazano na **listingu 1**, zaś kod odpowiedzialny za powiązanie zdarzenia przepełnienia licznika TCC5 z kanałem CH0 systemu zdarzeń na **listingu 2**. Dalej na **listingu 3** pokazano kod odpowiedzialny za inicjalizację przetwornika ADC mikrokontrolera. Jak widać, wybrano 12-bitowy

tryb SINGLE ENDED ze znakiem, sprzętowe uśrednianie wyników 16 pomiarów wraz ze sprzętowym przesunięciem wyniku o 4 bity w prawo (dla zwiększenia odstepu

sygnału od szumu), zewnętrzne źródło napięcia odniesienia podłączone do portu PD0 mikrokontrolera oraz jako wyzwalacz konwersji kanał CH0 systemu zdarzeń. Ponadto



Rysunek 1. Schemat ideowy urządzenia Volta



Rysunek 2. Schemat blokowy systemu akwizycji danych

Listing 1. Kod odpowiedzialny za inicjalizację układu czasowo-licznikowego TCC5

```
//Uruchomienie timera TCC5
//by przepełniał się 10 razy/sek
//sterowanie akwizycją danych
TCC5_CTRLB = TC_WGMODE_NORMAL_gc;
TCC5_PER = 24999; //10Hz @ 2MHz
TCC5_CTRLA = TC_CLKSEL_DIV8_gc;
```

Listing 2. Kod odpowiedzialny za powiązanie zdarzenia przepełnienia licznika TCC5 z kanałem CH0 systemu zdarzeń

```
//Kanał 0 systemu zdarzeń będzie
//przekazywał zdarzenie przepełnienia TCC5
EVSYS_CH0MUX = EVSYS_CHMUX_TCC5_OVF_gc;
```

Listing 3. Kod odpowiedzialny za inicjalizację przetwornika ADC mikrokontrolera Xmega

```
void initADC(void){
//ADC config: 12bit single ended, High current limit,
//External reference, gain = x1, trigger - event ch0
//ADC Enabled
ADCA_CTRLA = ADC_ENABLE_bm;
ADCA_CTRLB = ADC_CURRLIMIT_HIGH_gc|ADC_RESOLUTION_12BIT_gc|ADC_CONMODE_bm;
//50kps max sampling rate, signed 12-bit right-adjusted result
//External reference on PORT D
ADCA_REFCTRL = ADC_REFSEL_AREFD_gc;
//First event triggers channel conversion
ADCA_EVCTRL = ADC_EVSEL_0_gc|ADC_EVACT_CH0_gc;
//500kHz @ 2MHz
ADCA_PRESCALER = ADC_PRESCALER_DIV4_gc;
ADCA_CH0_CTRL = ADC_CH_GAIN_1X_gc|ADC_CH_INPUTMODE_SINGLEENDED_gc;
//pin PA7
ADCA_CH0_MUXCTRL = ADC_CH_MUXPOS_PIN7_gc;
//Interrupt on conversion complete
//High level
ADCA_CH0_INTCTRL = ADC_CH_INTMODE_COMPLETE_gc|ADC_CH_INTLVL_HI_gc;
//Averaged 16 samples and right shifted by 4
ADCA_CH0_AVGCTRL = (4<<4)|ADC_CH_SAMPNUM_16X_gc;
}
}
```

Listing 4. Kod funkcji obsługi przerwania przetwornika ADC odpowiedzialnej za akwizycję danych

```
//Makra akwizycji danych
#define DATA_ACQUISITION_ON TCC5_CTRLA = TC_CLKSEL_DIV8_gc
#define DATA_ACQUISITION_OFF TCC5_CTRLA = TC_CLKSEL_OFF_gc

ISR(ADCA_CH0_vect){
ADCready = 1;
//Wyłączenie akwizycji danych
DATA_ACQUISITION_OFF;
}
}
```

Listing 5. Kod odpowiedzialny za inicjalizację układów czasowo-licznikowych TCC4 i TCD5

```
//Uruchomienie timera TCC4 w trybie PWM typu Single Slope
//obsługa buzzera piezo
TCC4_CTRLB = TC_WGMODE_SINGLESLOPE_gc;
TCC4_PER = 249; //4kHz @ 2MHz
TCC4_CCC = 125; //Duty cycle 50%
//Output Compare enabled on OC4C pin
TCC4_CTRLA = TC_CCCMODE_COMP_gc;
PORTC_DIR |= PIN2_bm; //OC4C as output

//Uruchomienie timera TCD5 by przepełniał się 100 razy/sek
//obsługa klawiatury i timery programowe
TCD5_CTRLB = TC_WGMODE_NORMAL_gc;
TCD5_PER = 2499; //100Hz @ 2MHz
TCD5_CTRLA = TC_CLKSEL_DIV8_gc;
TCD5_INTCTRLA = TC_OVFINTLVL_MED_gc;
```

uruchomiono przerwanie po konwersji przetwornika ADC odpowiedzialne za akwizycję danych listingu 4.

Uważny Czytelnik zastanowi się zapewne, dlaczego wybrano tryb ze znakiem, skoro nie zamierzamy mierzyć napięć ujemnych względem masy. Takie ustawienie zmniejsza wynikową rozdzielczość przetwornika ADC z 12 na 11 bitów, gdyż jeden z bitów poświęcony jest na znak (+/-) wartości wynikowej. Otóż przetwornik ADC wbudowany w strukturę mikrokontrolera ATXmega8E5 w konfiguracji SINGLE ENDED i bez znaku (czyli w pełnej 12-bitowej rozdzielczości) jest w stanie mierzyć napięcia z zakresu $-\Delta V \dots V_{ref} - \Delta V$, gdzie ΔV jest offsetem wprowadzanym do pomiaru równym w przybliżeniu $0,05 \times V_{ref}$, co daje w naszym przypadku wartość 125 mV (gdzie $V_{ref} = 2,5$ V). Funkcjonalność tę wprowadzono głównie po to, by

mikrokontroler był w stanie w łatwy sposób wykrywać przejście mierzonego sygnału przez 0, co w założeniach miało uprościć implementację wszelkiego rodzaju falowników i innych układów sterujących. Finałnie sprowadza się do tego, że napięciu równemu 0 V (GND) nie odpowiada oczekiwana wartość wynikowa ADC równa 0, a wielkość offsetu zbliżona do bezwzględnej wartości w zakresie 200...205. Co więcej, wartość ta zależna jest od egzemplarza danego mikrokontrolera i należałoby ją wyznaczyć w procedurze kalibracji polegającej na zwarciu wejścia pomiarowego ADC do masy i pomiarze wartości wynikowej ADC. Kolejnym ograniczeniem tego trybu pracy jest maksymalny zakres pomiarowy przetwornika ADC, który w tym wypadku wynosi $0,95 \times V_{ref}$ (2,375 V), co po przeskalowaniu przez wejściowy dzielnik napięciowy

Wykaz elementów:

Rezystory: (SMD 0805)

R1, R2: 4,7 kΩ
R3: 11,3 kΩ 0,1%
R4: 215 kΩ 0,1%
R5: 1 kΩ

Kondensatory: (SMD 0805)

C1, C2, C4: 100 nF X7R
C3: 1 μF X7R

Półprzewodniki:

U1: ATXmega8E5-AU (TQFP32)
U2: TS9011SCX (SOT23)
U3: LM4040CYM3-2.5 (SOT23)
OLED: wyświetlacz OLED 128×32 px, 0,91", SSD1306, I2C, 38×12 mm

Pozostałe:

L1: dławik SMD 10 μH (SMD 0805)
PLUS, MINUS: mikroprzełącznik SMD typu DTSM31NB
BUZZ: przetwornik piezoelektryczny typu LPT9018BS-HL-03-4.0-12-R
VIN: AK500/2

odpowiada wejściowemu napięciu o wartości 47,5 V. Biorąc to wszystko pod uwagę, ostatecznie zrezygnowałem z pełnej rozdzielczości równej 12 bitów ograniczonej w wyżej przytoczony sposób, lecz zdecydowałem się na tryb SINGLE ENDED ze znakiem, który po pierwsze nie wymaga kalibracji, a po drugie zmniejsza wynikową rozdzielczość pomiaru do 11 bitów.

Zakończeniu konwersji danych ADC towarzyszy zatrzymanie akwizycji danych (wyłączenie timera TCC5) oraz ustawienie flagi *ADCready*, dzięki czemu możliwe jest przetworzenie danych (rejestr ADCA.CH0RES mikrokontrolera) przez program główny aplikacji. Tyle w kwestii wspomnianej akwizycji danych.

Kolejne podsystemy mikrokontrolera, które używane są w aplikacji urządzenia Volta, to dwa dodatkowe układy czasowo-licznikowe, a mianowicie TCC4 i TCD5. Pierwszy ze wspomnianych timerów (TCC4) pracuje w trybie PWM typu Single Slope i odpowiada za obsługę buzzera piezo, generując przebieg prostokątny o częstotliwości 4 kHz na wyprowadzeniu PC2 mikrokontrolera, zaś drugi (TCD5) pracuje w trybie normalnym skonfigurowany tak, by przepełniał się 100 razy/s i odpowiedzialny jest za obsługę klawiatury (debouncing), jak i timery programowe. Kod odpowiedzialny za inicjalizację obu układów czasowo-licznikowych pokazano na listingu 5.

REKLAMA

KAMAMI www.kamami.pl

wszystko o mikrokontrolerach

STM32

sprawdź naszą ofertę na **KSIĄŻKI** i **eBOOKI**

Listing 6. Funkcja odpowiedzialna za inicjalizację interfejsu TWI mikrokontrolera Xmega

```
void TWIinit(void){
    //Ustawienie kierunków portów SDA i SCL
    SDA_DIR_REG |= (1<<SDA_PIN);
    SCL_DIR_REG |= (1<<SCL_PIN);
    //Ustawienie częstotliwości zegara SCL = 200kHz
    //FTWI = F_CPU/2*(5+BAUD)
    TWIC.MASTER.BAUD = 0;
    //Odblokowanie modułu TWI master
    TWIC.MASTER.CTRLA = TWI_MASTER_ENABLE_bm;
    //Początkowe wykasowanie wszystkich flag
    TWIC.MASTER.STATUS |= TWI_MASTER_RIF_bm|TWI_MASTER_WIF_bm|TWI_MASTER_ARBLOST_bm|TWI_MASTER_BUSERR_bm;
    //Ustawienie stanu wyjściowego interfejsu TWI
    TWIC.MASTER.STATUS |= TWI_MASTER_BUSSTATE_IDLE_gc;
}
```

W tym miejscu warto poświęcić kilka słów konfiguracji timera TCD5 oraz przerwaniu od przepelnienia licznika, które jest wywoływane 100 razy na sekundę i odpowiada za obsługę klawiatury urządzenia. I właśnie tutaj kryje się mały haczyk, na który i ja dałem się złapać. Otóż, w przypadku mikrokontrolera Xmega, wywołaniu funkcji obsługi przerwania od przepelnienia licznika TCD5 nie towarzyszy automatyczne skasowanie flagi tego przerwania (OVFIF w rejestrze INTFLAGS), jak miało to miejsce w przypadku starszych mikrokontrolerów AVR (rodziny Mega i Tiny). Flagę tę musimy skasować programowo (`TCD5.INTFLAGS = TC5_OVFIF_bm`), w innym wypadku przerwanie od przepelnienia będzie wywoływane non stop, uniemożliwiając w zasadzie poprawne działanie urządzenia. To dość istotna właściwość timerów mikrokontrolera Xmega serii E, którą warto mieć na uwadze.

Ostatnim podsystemem mikrokontrolera, jaki wykorzystuje urządzenie Volta, jest interfejs szeregowy transmisji danych TWI będący odpowiednikiem interfejsu I²C. Interfejs ten jest używany do współpracy z wyświetlaczem OLED o rozdzielczości 128×32 piksele, stanowiącym element graficznego interfejsu użytkownika. Wspomniany interfejs występuje w większości mikrokontrolerów AVR, zarówno tych starszych, jak i najnowszych konstrukcji z rodziny ATXmega, jednak w wykonaniu tych ostatnich został rozbudowany w stosunku do jego pierwszych implementacji oferując dodatkowe funkcjonalności. Przede wszystkim TWI wspiera duże prędkości zegara (aż do 1 MHz), tryb multi-master (z obsługą arbitrażu), możliwość pracy w standardzie SMBus, adresowanie 7- i 10-bitowe, wybudzenie procesora ze stanu uśpienia i wiele, wiele innych. Warto zauważyć, że dokumentacja w tym zakresie jest bardzo obszerna i drobiazgowa, więc kompletny opis interfejsu wykraczałby poza ramy niniejszego artykułu. Co więcej, producent dostarcza gotowy sterownik do obsługi sprzętu TWI (dość rozbudowany i korzystający z przerwań) oraz notę aplikacyjną o oznaczeniu „AVR1308: Using the XMEGA TWI”. My, z racji prostoty urządzenia, posłużymy się elementarnymi funkcjami narzędziowymi wykorzystującymi tzw. *pooling*.

Zacniemy od funkcji inicjującej interfejs TWI mikrokontrolera Xmega, która

przedstawiono na **listingu 6**. Dalej, na **listingu 7** pokazano funkcję odpowiedzialną za wygenerowanie sygnału Start interfejsu TWI i przesłanie adresu układu Slave. Prostą funkcję narzędziową, której zadaniem jest wygenerowanie sygnału Stop magistrali TWI i tym samym zakończenie transmisji danych, pokazano na **listingu 8**. Na **listingach 9 i 10** pokazano kolejne, dwie proste funkcje umożliwiające wysłanie i odebranie bajta danych na magistrali TWI. Funkcja wysyłająca bajt jako rezultat swojego działania zwraca wartość bitu potwierdzenia (ACK) po stronie odbiornika danych (w tym przypadku układu Slave), zaś funkcja

pozwalająca na odbiór bajta przyjmuje jako argument wywołania wartość bitu potwierdzenia po stronie odbiornika danych (w tym przypadku układu Master).

Urządzenie wyposażono w niewielki, ale efektowny wyświetlacz OLED. Powodem wybrania tego typu wyświetlacza jest jego doskonała widoczność nawet w świetle słonecznym i bardzo atrakcyjna cena. Zastosowany moduł 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

Listing 7. Funkcja odpowiedzialna za wygenerowanie sygnału Start interfejsu TWI i przesłanie adresu układu Slave

```
uint8_t TWIstart(uint8_t slaveAddress){
    //Inicjujemy wygenerowanie sygnału Start
    //i przesłanie adresu układu Slave
    TWIC.MASTER.ADDR = slaveAddress;

    //Czekamy na przesłanie adresu układu Slave
    while(!(TWIC.MASTER.STATUS & (TWI_MASTER_WIF_bm)));

    //Zwracamy NACK, jeśli odebrano sygnał NACK,
    //przebrano arbitraż lub też wystąpił inny błąd magistrali TWI
    if(TWIC.MASTER.STATUS &
        (TWI_MASTER_RXACK_bm|
         TWI_MASTER_ARBLOST_bm|
         TWI_MASTER_BUSERR_bm)) return NACK;
    else return ACK;
}
```

Listing 8. Kod funkcji odpowiedzialnej za wygenerowanie sygnału Stop magistrali TWI i tym samym zakończenie transmisji danych

```
void TWIstop(void) {
    //Inicjujemy wygenerowanie sygnału Stop poprzedzonego NACK
    TWIC.MASTER.CTRLA = TWI_MASTER_ACKACT_bm|TWI_MASTER_CMD_STOP_gc;
}
```

Listing 9. Funkcja odpowiedzialna za wysłanie bajta danych przy użyciu magistrali TWI

```
uint8_t TWIwriteByte(uint8_t Data) {
    //Inicjujemy przesłanie bajta danych
    TWIC.MASTER.DATA = Data;
    //Czekamy na przesłanie bajta danych
    while(!(TWIC.MASTER.STATUS & (TWI_MASTER_WIF_bm)));
    //Zwracamy NACK, jeśli odebrano sygnał NACK,
    //przebrano arbitraż lub też wystąpił inny błąd magistrali TWI
    if(TWIC.MASTER.STATUS &
        (TWI_MASTER_RXACK_bm|
         TWI_MASTER_ARBLOST_bm|
         TWI_MASTER_BUSERR_bm)) return NACK;
    else return ACK;
}
```

Listing 10. Funkcja odpowiedzialna za odbiór bajta danych przy użyciu magistrali TWI

```
uint8_t TWIreadByte(uint8_t ACKNACK) {
    uint8_t Data;
    //Czekamy na odbiór bajta danych
    while(!(TWIC.MASTER.STATUS & TWI_MASTER_RIF_bm));
    //Odczytujemy przesłany bajt danych
    Data = TWIC.MASTER.DATA;
    //Wysyłamy sygnał ACK/NACK
    if (ACKNACK == ACK) TWIC.MASTER.CTRLA = TWI_MASTER_CMD_RECVTRANS_gc;
    return Data;
}
```

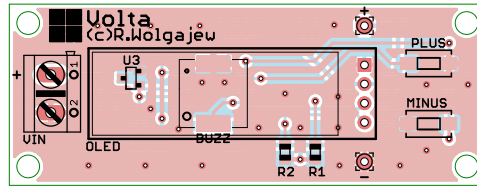
rozwiązanie, gdyż wymaga zaangażowania wyłącznie 2 wyprowadzeń mikrokontrolera do przeprowadzenia transmisji z modulem wyświetlacza, lecz niepozbawione wad, gdzie jedną z nich jest niewielka prędkość transmisji wpływająca na częstość odświeżania ekranu. Po szczegóły implementacyjne dotyczące tegoż peryferium odsyłam do mojego artykułu opisującego urządzenie USBtester opublikowanego w EP 05/19, gdzie drobiazgowo opisano zagadnienia programowe związane z obsługą tego rodzaju wyświetlacza OLED.

Montaż i uruchomienie

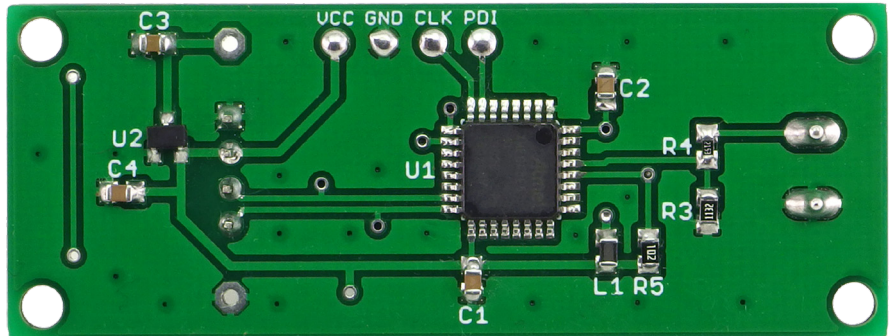
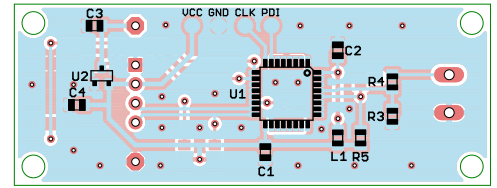
Schemat montażowy urządzenia pokazano na **rysunku 3**. Zaprojektowano niewielką, dwustronną płytkę drukowaną, która swoim rozmiarem zbliżona jest do zastosowanego wyświetlacza OLED, udostępniając dodatkowo niezbędne otwory montażowe. Warto również podkreślić, że dla zminimalizowania rozmiaru obwodu drukowanego przewidziano tutaj montaż elementów po obu stronach laminatu. Aplikację urządzenia rozpoczynamy od przylutowania mikrokontrolera (warstwa BOTTOM). Dalej lutujemy pozostałe elementy półprzewodnikowe, potem rezystory i kondensatory oraz inne elementy bierne. Na warstwie TOP montujemy układ U3, następnie buzzer i pozostałe elementy bierne, a na końcu mikroprzełączniki PLUS i MINUS oraz gniazdo podłączeniowe VIN.

Z uwagi na zagęszczenie wyprowadzeń układów scalonych przed pierwszym podłączeniem układu do zasilania należy jeszcze raz sprawdzić jakość wykonanych połączeń, aby nie dopuścić do ewentualnych zwarc. Wspomniana kontrola będzie znacznie łatwiejsza, jeśli zmontowaną płytkę przemyjemy alkoholem izopropylowym w celu wyłukania nadmiaru kalafonii lutowniczej.

Do tak przygotowanej płytki montujemy wyświetlacz OLED, zwyczajnie lutując jego wyprowadzenia w przewidziane do tego celu



Rysunek 3. Schemat montażowy urządzenia



Fotografia 1. Zmontowana płytkę, widok od strony BOTTOM

pola lutownicze (należy sprawdzić polaryzację zasilania), gdyż połączenia elektryczne zapewniają mu jednocześnie wystarczająco stabilny montaż mechaniczny. Poprawnie zmontowany układ powinien działać zaraz po włączeniu zasilania. Na **fotografii 2** pokazano zmontowane urządzenie od strony BOTTOM.

Obsługa

Po włączeniu zasilania (zaciski +/- na obwodzie drukowanym) urządzenie pracuje w trybie pokazywania wartości mierzonego napięcia (rozdzielczość 20 mV), przy czym ekran w tym trybie odświeżany jest 10 razy na sekundę. Każdorazowe wciśnięcie klawiszy PLUS/MINUS powoduje chwilowe (2 s) przełączenie urządzenia w tryb pokazywania wartości alarmu dla mierzonego napięcia, co symbolizowane jest poprzez wyświetlenie symbolu dzwoneczka po lewej stronie wartości napięcia. W trybie tym krótkie naciśnięcie wspomnianych klawiszy powoduje każdorazową regulację wartości napięcia



Rysunek 4. Wygląd interfejsu użytkownika

alarmowania o 10 mV, długie naciśnięcie o 100 mV, zaś naciśnięcie i przytrzymanie o 1 V. Ustawiona wartość progu alarmowania zapamiętywana jest każdorazowo w nieulotnej pamięci EEPROM mikrokontrolera i wczytywana podczas włączania urządzenia. Przekroczenie ustawionej wartości alarmowania przez mierzone napięcia powoduje generowanie pojedynczych dźwięków poprzez wbudowany buzzer piezoelektryczny.

Podczas włączania urządzenia uruchamiany jest proces autokalibracji przetwornika ADC, w związku z tym nie należy w tym czasie podłączać jakichkolwiek obwodów pomiarowych do wejścia VIN urządzenia. Wygląd graficznego interfejsu użytkownika urządzenia Volta pokazano na **rysunku 4**.

Robert Wołgajew, EP

REKLAMA

ELEKTRONIKA
PRAKTYCZNA

MATERIAŁY
DODATKOWE



MEDIA

Aby skorzystać z materiałów dodatkowych dołączonych do numeru, należy:

1. Wejść na stronę www.media.avt.pl,
2. Zarejestrować się lub zalogować,
3. Wybrać wydanie „Elektroniki Praktycznej”, które ma trafić do biblioteki osobistej,
4. Odpowiedzieć na proste pytanie dotyczące bieżącego numeru,
5. Pobrać pliki.