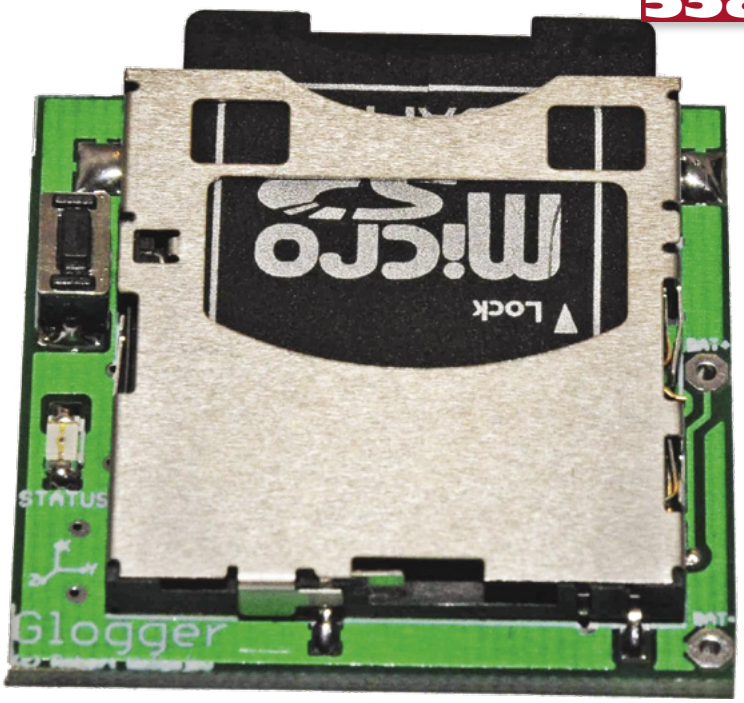
gLogger

3-osiowy rejestrator przyśpieszenia

AVT 5387

Współcześnie dołączenie do mikrokontrolera AVR karty pamięci SD i układu do pomiaru przyspieszenia nie stanowi problemu. Przykładem takiego połączenia jest prezentowany gLogger – energooszczędny system do rejestrowania przyspieszenia statycznego (siły przyciągania ziemskiego) i dynamicznego, rejestrujący wartości przyśpieszenia w trzech osiach na wbudowanej karcie pamięci SD. ***Rekomendacje:*** *miernik może przydać się w układach nawigacji inercyjnej lub pasjonatom motoryzacji, do pomiaru przyśpieszenia pojazdu.*



Jeszcze do niedawna budowa systemu do pomiaru nieelektrycznych wielkości fizycznych pozostawała w sferze marzeń, głównie za sprawą niezbędnego wyposażenia, bez którego nie sposób było wykonać precyzyjne elementy układów pomiarowych. Na szczęście, rozwój technologii MEMS i jej ekspansja na coraz to nowe płaszczyzny zastosowań (kompleksowe systemy pomiarowe, przetworniki audio, kompletne medyczne systemy diagnostyczne) zrewolucjonizował tę gałąź elektroniki, co umożliwiło z kolei implementację tego typu pomiarów na prostych platformach sprzętowych.

Schemat proponowanego rozwiązania rejestratora przyśpieszenia pokazano na **rysunku 1**. Jego „sercem" jest mikrokontroler ATmega32A taktowany za pomocą wewnętrznego oscylatora (8 MHz). Ten układ realizuje obsługę scalonego, 3-osiowego czujnika przyspieszenia ADXL343 z użyciem sprzętowego interfejsu szeregowego TWI (kompatybilnego z I²C). Karta pamięci SD jest obsługiwana za pomocą sprzętowego interfejsu SPI oraz nieskomplikowanego interfejsu użytkownika złożonego z jednej diody LED i jednego przycisku typu microswitch.

Akcelerometr ADXL343

Pomiar przyspieszenia jest możliwy dzięki zastosowaniu scalonego akcelerometru 3-osiowego ADXL343 z wyjściem cyfrowym. Akcelerometr jest elementem o doskonałych parametrach elektrycznych. Ponadto, dzięki wyposażeniu go w interfejsy I²C oraz SPI, jego dołączenie do systemu pomiarowego jest bardzo łatwe. Dzięki bogatemu wyposażeniu czujnika nie ma potrzeby stosowania wysokostabilnych przetworników A/C w celu uzyskania wartości pomiarowych.

Akcelerometr ma możliwość elastycznego konfigurowania. Charakteryzuje się następującymi wybranymi parametrami użytkowymi:

- pomiar przyspieszenia statycznego oraz dynamicznego w 3 osiach z maksymalną rozdzielczością 13-bitów,
- rozdzielczość jest konfigurowalna (3,9 mg/LSB), dzięki czemu akcelerometr może być z powodzeniem stosowany w układach pomiaru nachylenia,
- szeroki zakres napięcia zasilającego: 2,0…3,6 V (interfejs I/O układu od 1,7 V),
- konfigurowalny zakres pomiarowy: ±2,0 g; 4,0 g; 8,0 g; 16,0 g;
- szeroki zakres częstotliwości pomiaru: 0,10…3200 Hz,
- mały pobór prądu: 0,04–0,15 mA,
- wbudowane, konfigurowalne mechanizmy obsługi zdarzeń, takich jak: stuknięcie (*tap*), podwójne stuknięcie (*double tap*), bezruch, ruch (aktywność), spadek swobodny,
- 2 niezależne, konfigurowalne wyprowadzenia przerwań sprzętowych generowanych przy wystąpieniu zdarzeń,
- wbudowany bufor FIFO o pojemności 32 pomiarów (32×6 bajtów),
- wbudowany tryb samokontroli (*selftest*) pozwalający na sprawdzenie funkcjonalności struktury elektronicznej i mechanicznej akcelerometru,
- możliwość automatycznej kompensacji offsetu pomiarowego.

Wygląd obudowy układu pokazano na **rysunku 2**, natomiast w **tabeli 1** umieszczono opis wyprowadzeń akcelerometru ADXL343.

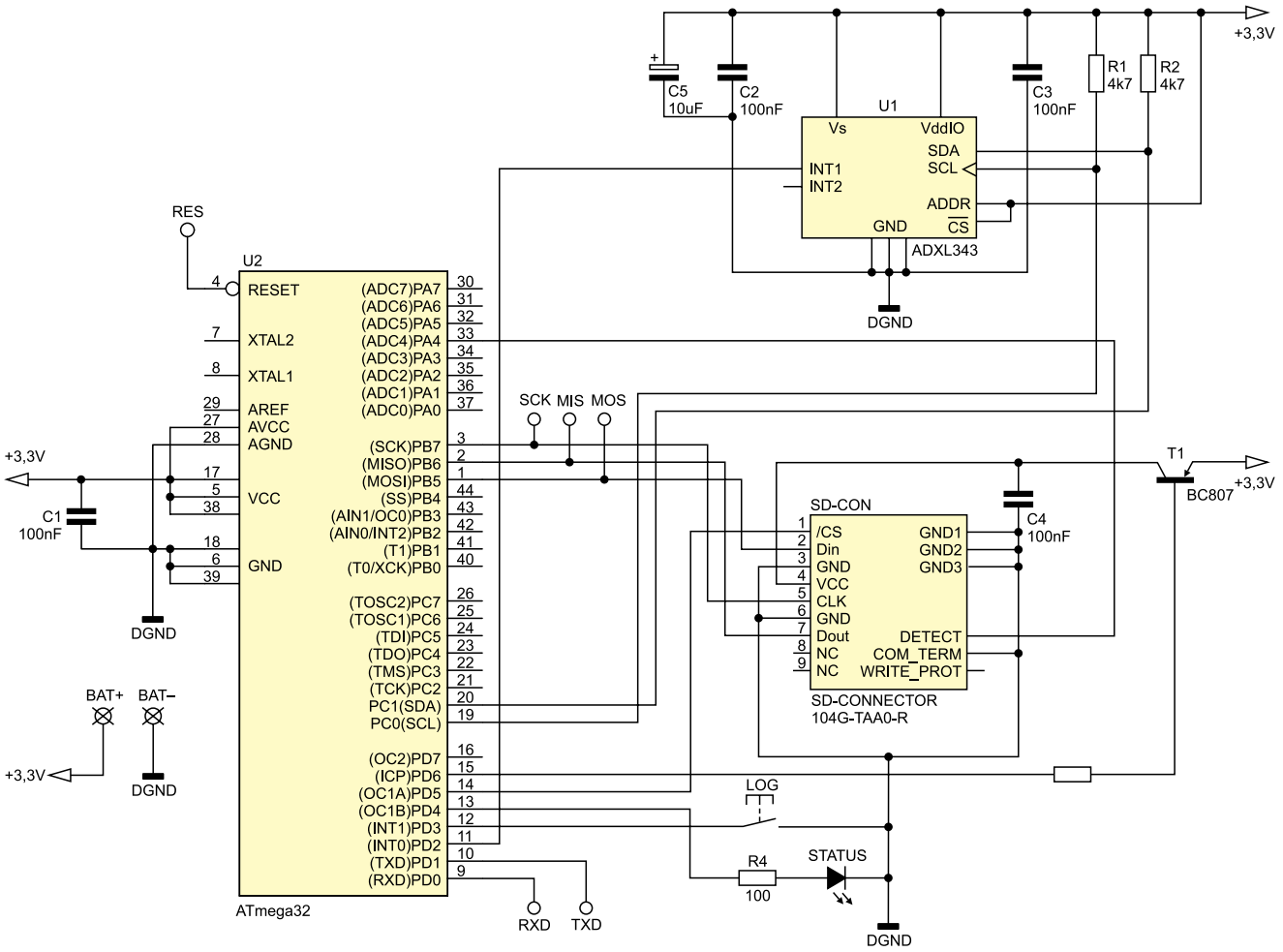
W ofercie AVT*
AVT-5387 A

Podstawowe informacje:
- Napięcie zasilania: 3,3 V DC.
- Prąd obciążenia (stan bezczynności/zapis na kartę SD/inicjalizacja): 700 µA/25 mA/34 mA (zależny od rodzaju i producenta karty pamięci).
- Zakres pomiarowy: ±16g.
- Rozdzielczość pomiaru przyspieszenia: 13 bitów (3.9 mg/LSB).
- Błąd liniowości: 0,5%.
- Dokładność pomiaru przyspieszenia: 1%.
- Interwał pomiarowy: 80 ms.

Dodatkowe materiały na CD/FTP:
ftp://ep.com.pl, user: 63048, pass: 632vmey5
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

Projekty pokrewne na CD/FTP:
(wymienione artykuły są w całości dostępne na CD)
AVT-5244 GPS-owy rejestrator trasy (EP 7/2010)
AVT-5223 Kieszonkowy akcelerometr (EP 2/2010)
Projekt 132 Miernik przyśpieszenia (EP 8/2005)
--- Elektroniczny miernik przyśpieszenia (EP 8/1998)

* Uwaga:
Zestawy AVT mogą występować w następujących wersjach:
AVT xxxx UK – to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.
AVT xxxx A – płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.
AVT xxxx A+ – płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.
AVT xxxx B – płytka drukowana (lub płytki) oraz komplet elementów wymieniony w załączniku pdf
AVT xxxx C – to nic innego jak zmontowany zestaw B, czyli elementy wlutowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf
AVT xxxx CD – oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)
Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). **http://sklep.avt.pl**

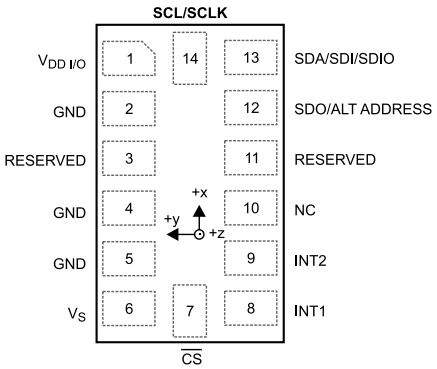


Rysunek 1. Schemat ideowy układu gLogger

Na **rysunku 3** przedstawiono odpowiedź układu ADXL343 w zależności od położenia jego obudowy w stosunku do współrzędnych XYZ. Już z pobieżnej lektury wymienionych parametrów użytkowych wynika, że producent akcelerometru wyposażył go w wiele przydatnych, przemyślanych rozwiązań sprzętowych i funkcjonalnych. Pozwalają one na łatwe użycie sensora w docelowym systemie mikroprocesorowym i zapewniają elastyczność aplikacji. Jednym z takich rozwiązań jest możliwość generowania przerwań sprzętowych po wystąpieniu zaprogramowanego zdarzenia. Istnieje przy tym możliwość wyboru wyprowadzenia akcelerometru, które ma być użyte do sygnalizowania przerwania (INT1 lub INT2) oraz określenia jego poziomu aktywnego (niski lub wysoki). Do zdarzeń, które mogą generować przerwania należą:

- gotowość danych pomiarowych (*DATA_READY*),
- pojedyncze stuknięcie (*TAP*), przy czym odpowiednie rejestry przechowują informację na temat osi, dla której wykryto wystąpienie zdarzenia,
- zdarzenie podwójnego stuknięcia (*DOUBLE_TAP*) z danymi jak wyżej,
- ruch urządzenia (*ACTIVITY*),
- bezruch urządzenia (*INACTIVITY*),
- swobodny upadek (*FREE_FALL*),
- zajętość bufora FIFO (*WATERMARK*), które to zdarzenie zachodzi, gdy liczba danych pomiarowych w buforze danych FIFO (jeśli jest używany) jest równa zdefiniowanej wartości.

Wszystkie zdarzenia, poza zdarzeniem *DATA_READY*, mają dodatkowe, związane z nimi rejestry konfiguracyjne, dzięki którym



Rysunek 2. Wygląd obudowy akcelerometru ADXL343 (widok z góry)

Tabela 1. Opis wyprowadzeń akcelerometru ADXL343

Pin	Nazwa	Opis
1	VDDI/O	Napięcie zasilania interfejsu I/O akcelerometru
2	GND	Masa
3	RESERVED	Zarezerwowany (nie podłączać)
4	GND	Masa
5	GND	Masa
6	VS	Napięcie zasilania akcelerometru
7	CS	CS dla interfejsu SPI (podłączyć do VS dla interfejsu I^2C)
8	INT1	Wyjście przerwania INT1 (konfigurowalne)
9	INT2	Wyjście przerwania INT2 (konfigurowalne)
10	NC	Niewykorzystane (nie podłączać)
11	RESERVED	Zarezerwowany (nie podłączać)
12	SDO/ALT ADDRESS	SDO dla interfejsu SPI 4-przewodowego lub wybór adresu dla interfejsu I^2C
13	SDA/SDI/SDIO	SDI dla interfejsu SPI 4-przewodowego, SDIO dla interfejsu SPI 3-przewodowego lub SDA dla interfejsu I^2C
14	SCL/SCLK	Sygnał zegarowy dla interfejsu SPI i I^2C

można określić parametry zdarzenia. Ponadto, dzięki wbudowanemu buforowi danych FIFO (możliwość pracy w trybach Bypass, FIFO, Stream i Trigger) oraz mechanizmowi *WATERMARK*, możliwe stało się odciążenie procesora z ciągłego nadzorowania stanu ADXL343, zaoszczędzenie mocy obliczeniowej i ograniczenie liczby przesyłanych danych.

Zasada działania

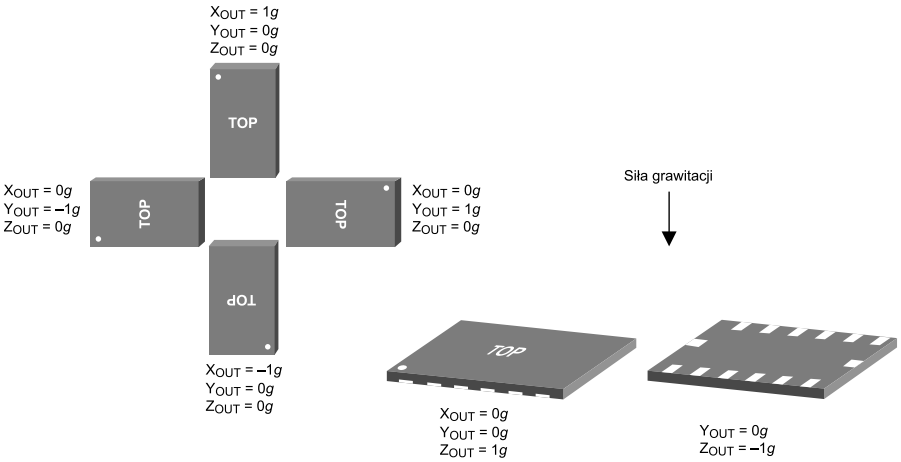
Akcelerometr ADXL343 jest obsługiwany przy użyciu wbudowanego w mikrokontroler ATmega32 interfejsu TWI pracującego z częstotliwością sygnału zegarowego 200 kHz. Tuż po włączeniu zasilania akcelerometr zostaje skonfigurowany do pracy z częstotliwością wykonywania pomiarów równą 12,5 Hz, jest wybierany zakres pomiaru 16g, pełna rozdzielczość pomiarowa (13-bitów) oraz aktywowane przerwanie od zdarzenia DATA_READY (gotowość danych pomiarowych) z aktywnym poziomem niskim (co oznacza, iż w stanie bezczynności na wyjściu INT1 akcelerometru występuje poziom wysoki). Następnie akcelerometr jest wprowadzany w tryb Standby (nie wykonuje pomiarów, pobór prądu rzędu 0,1 μA), oczekując na rozkaz inicjujący pomiary.

Mikrokontroler, po przeprowadzeniu niezbędnych czynności dotyczących obsługi karty pamięci SD, konfiguruje własne przerwania zewnętrzne INT0 i INT1 jako przerwania wyzwalane poziomem niskim, zezwala na obsługę przerwania INT1 (odpowiedzialnego za obsługę przycisku LOG) oraz przechodzi w tryb Power-Down (pobór prądu rzędu 0,4 μA), z którego może zostać wybudzony poprzez obsługę przycisku LOG (na tę chwilę). Następnie, każde przyciśnięcie przycisku LOG wybudza mikrokontroler ze stanu PowerDown i powoduje wykonanie procedury obsługi przerwania INT1, w której następuje aktywacja/deaktywacja procesu pomiarowego i ponowne uśpienie mikrokontrolera (już w pętli głównej programu obsługi).

Aktywacja procesu pomiarowego polega na zezwoleniu na przerwanie INT0 oraz wy-

Listing 1. Procedura obsługi przerwania INT0 odpowiedzialna za odczyt danych przyspieszenia.

```
,Ta procedura zachodzi, gdy akcelerometr ma gotowy komplet danych do odczytania - wtedy zeruje wyprowadzenie INT1
i utrzymuje ten poziom do czasu odczytania wszystkich danych. Odczytanie danych z pamięci akcelerometru powoduje
ustawienie wyprowadzenia INT1.
,Zmienne przyspieszenia (13 bitów):  S X11...X8| X7...X0 -> zakres +- 4095 -> 3.9g/LSB
Read_acceleration:
  Twcr = &B10100100  ,Wysłanie START: TWINT=1, TWSTA=1, TWEN=1
,Czekamy, aż interfejs TWI wykona Start i ustawi flagę TWINT
  Bitwait Twcr.twint , Set
,Do rejestru danych TWI wpisujemy adres akcelerometru w trybie zapisu
  Twdr = Accel_write_addr
  Twcr = &B10000100
,Inicjujemy proces wysłania bajta danych: TWINT=1, TWSTA=0, TWEN=1. ,Czekamy, aż TWI wyśle Adres i ustawi flagę TWINT
  Bitwait Twcr.twint , Set
,Do rejestru danych TWI wpisujemy adres rejestru X0
  Twdr = Data_x0_reg
,Inicjujemy proces wysłania bajta danych: TWINT=1, TWSTA=0, TWEN=1
  Twcr = &B10000100
,Czekamy, aż interfejs TWI wyśle Adres i ustawi flagę TWINT
  Bitwait Twcr.twint , Set
,Inicjujemy wysłanie sygnału RESTART: TWINT=1, TWSTA=1, TWEN=1
  Twcr = &B10100100
,Czekamy, aż interfejs TWI wykona Start i ustawi flagę TWINT
  Bitwait Twcr.twint , Set
,Do rejestru danych TWI wpisujemy adres akcelerometru w trybie odczytu
  Twdr = Accel_read_addr
,Inicjujemy wysłanie bajta: TWINT=1, TWEN=1, TWSTA=0
  Twcr = &B10000100
,Czekamy, aż TWI wyśle Adres i ustawi flagę TWINT
  Bitwait Twcr.twint , Set
,Inicjujemy odbiór bajta - potwierdzamy ACK
  Twcr = &B11000100
,Czekamy na odbiór bajtu sygnalizowany flagą TWINT
  Bitwait Twcr.twint , Set
  Byte0 = Twdr       ,X0
,Inicjujemy odbiór bajta - potwierdzamy ACK
  Twcr = &B11000100
,Czekamy na odbiór bajtu sygnalizowany flagą TWINT
  Bitwait Twcr.twint , Set
  Acc_x = Twdr And &B00011111  ,X1
  Shift Acc_x , Left , 8
,13-bitowa wartość przyspieszenia w osi X w kodzie U2
  Acc_x = Acc_x Or Byte0
,Inicjujemy odbiór bajtu - potwierdzamy ACK
  Twcr = &B11000100
,Czekamy na odbiór bajtu sygnalizowany ustawieniem TWINT
  Bitwait Twcr.twint , Set
  Byte0 = Twdr       ,Y0
,Inicjujemy odbiór bajta - potwierdzamy ACK
  Twcr = &B11000100
,Czekamy na odbiór bajtu sygnalizowany ustawieniem TWINT
  Bitwait Twcr.twint , Set
 Acc_y = Twdr And &B00011111  ,Y1
 Shift Acc_y , Left , 8
,13-bitowa wartość przyspieszenia w osi Y w kodzie U2
 Acc_y = Acc_y Or Byte0
,Inicjujemy odbiór bajtu - potwierdzamy ACK
  Twcr = &B11000100
,Czekamy na odbiór bajtu sygnalizowany ustawieniem TWINT
  Bitwait Twcr.twint , Set
  Byte0 = Twdr  ,Z0
,Inicjujemy odbiór bajtu - bez potwierdzania, bo to ostatni bajt
  Twcr = &B10000100
  Bitwait Twcr.twint , Set
  Acc_z = Twdr And &B00011111  ,Z1
  Shift Acc_z , Left , 8
,13-bitowa wartość przyspieszenia w osi Z w kodzie U2
  Acc_z = Acc_z Or Byte0
,Wysłanie sygnału Stop: TWINT=1, TWSTO=1 i TWEN=1
  Twcr = &B10010100
,Konwersja na tekst
  Strvalue = Str(acc_x) + „, „ + Str(acc_y) + „, „ + Str(acc_z)
,Zapamiętanie danych na karcie SD - wartości przyspieszeń w kodzie U2
,Konwersja na kod dziesiętny i mnożenie przez 3,9mg
  Open „log.txt” For Append As #1  ,Otwieramy plik
  Print #1 , Strvalue
  Close #1  ,Zamykamy plik
Return
```



Rysunek 3. Odpowiedź układu ADXL343 w zależności od położenia jego obudowy w stosunku do osi XYZ

Wykaz elementów
Rezystory: (SMD 0603)
R1, R2: 4,7 kΩ
R3: 2,2 kΩ
R4: 100 Ω
Kondensatory: (SMD 0603)
C1…C4: 100 nF
C5: 10 µF/10 V (typ B, EIA 3528-21)
Półprzewodniki:
U1: ADXL343 (obudowa LGA-14)
U2: ATmega32A (obudowa TQFP-44)
STATUS: dioda LED, czerwona, SMD1206
T1: BC807 lub podobny (SOT-23)
Inne:
SD-CON: gniazdo karty SD typu push-push o oznaczeniu 104G-TAA0-R (ATTEND)
LOG: microswitch SMD DTSM31

Ustawienia fusebits:
CKSEL3...0: 0100
SUT1...0: 10
CKOPT: 1
JTAGEN: 1
BODEN: 1
SPIEN: 0
OCDEN: 1
BOOTRST: 1

słaniu rozkazu inicjującego ustawiczny proces pomiarowy układu ADXL343. Zdarzenie gotowości danych uruchomionego procesu pomiarowego, które w naszym przypadku zachodzi co 80 ms (12,5 Hz) powoduje wyzerowanie wyprowadzenia INT1 akcelerometru, które to inicjuje obsługę przerwania INT0 mikrokontrolera. Wywołanie procedury obsługi przerwania powoduje wyjście mikrokontrolera z trybu PowerDown, odczytanie zmierzonych wartości przyśpieszenia oraz ich zapisanie na karcie pamięci SD w popularnym formacie CSV. Zapamiętywane są „surowe", binarne wartości przyspieszenia w kodzie U2. Po wykonaniu tych czynności mikrokontroler przechodzi ponownie do trybu PowerDown aż do wystąpienia kolejnego przerwania tego typu (lub przerwania INT1).

Procedurę obsługi przerwania INT0 zamieszczono na **listingu 1**. Jest ona zoptymalizowana pod kątem szybkości wykonania, co jednocześnie czyni ją nieco dłuższą, jeśli chodzi o wygenerowany kod.

Kilka słów uwagi należy się także obsłudze karty pamięci SD, ponieważ jest to dość skomplikowane zagadnienie z punktu widzenia programisty. Uważny Czytelnik z pewnością już zauważył, iż zasoby sprzętowe mikrokontrolera zostały użyte w naszym systemie tak naprawdę w niewielkim stopniu, jednak należy mieć na uwadze, iż system ten rejestruje dane pomiarowe na przewidzianej do tego celu karcie pamięci SD sformatowanej dla systemu plików FAT (16/32) co pociąga za sobą stosowanie „pamięciożernych" bibliotek, a więc i mikrokontrolerów zaopatrzonych w odpowiednią pamięć RAM. Łatwa implementacja tej funkcjonalności jest możliwa dzięki użyciu biblioteki AVR-DOS dostarczanej z kompilatorem Bascom AVR, której autorem jest Franz Vögel. Umożliwia ona obsługę kart pamięci z systemami plików FAT16 i FAT32. Korzystanie z biblioteki nie wiąże się z koniecznością ponoszenia dodatkowych kosztów – dla celów niekomercyjnych jest ona całkowicie bezpłatna.

Co oczywiste, używana karta pamięci powinna być odpowiednio sformatowana. Dodatkowo, w konstrukcji urządzenia wykorzystano specjalne wyprowadzenie złącza karty SD oznaczone jako DETECT a służące do sprawdzenia jej obecności w złączu (detekcja wyłącznie mechaniczna). Wspomniane sprawdzenie dokonywane jest wyłącznie w trakcie włączania urządzenia a co za tym idzie, nie należy wyjmować karty SD w trakcie jego pracy. Z resztą ta ostatnia kwestia ma przede wszystkim na uwadze bezpieczeństwo pracy samej karty i kontrolera w niej zawartego (o czym dobitnie przekonałem się uszkadzając jedną z kart w trakcie testów urządzenia). Procedurę inicjująca sterownik karty jak i mechanizm obsługi systemu plików przedstawiono na listingu **List.2**.

Po wykonaniu pomyślnej inicjalizacji karty SD i systemu plików FAT następuje wyszukiwanie pliku danych o nazwie „log.txt" w katalogu głównym karty SD. Jeśli operacja ta zakończy się niepowodzeniem, to plik zostanie utworzony. Brak karty w gnieździe SD, błąd inicjalizacji sterownika karty czy też niepoprawna inicjalizacja systemu plików (np. spowodowana niesformatowaniem karty SD) zostaną zasygnalizowane za pomocą diody LED.

Przyciśnięcie przycisku LOG (włączenie/wyłączenie rejestracji danych) jest sygnalizowane diodą LED. Pojedyncze mrugnięcie oznacza uaktywnienie rejestracji danych, natomiast podwójne – jej wyłączenie. Domyślnie, po

Listing 2. Procedura inicjująca sterownik karty SD jak i mechanizm obsługi systemu plików.

```
,Jeśli włożono kartę SD (sprawdzamy odpowiedni styk złącza karty) to
inicjujemy sterownik karty i sprawdzamy czy zgłoszono błąd
If Sd_card_plugged = 0 Then
  $include „Config_MMC.bas"
  Gbdriveerror = Driveinit()
  While Gbdriveerror <> 0
    Set Vcc  ,Wyłączenie zasilania karty SD - OFF
    Waitms 10
    Reset Vcc  ,Włączenie zasilania karty SD - ON
    Waitms 10
    Gbdriveerror = Driveinit()
    Call Blinking(2)  ,Błąd inicjacji sterownika karty
    Waitms 500
  Wend
,Jeśli inicjacja sterownika karty przebiegła prawidłowo to inicjujemy system
plików
  If Gbdriveerror = 0 Then
    $include „Config_AVR-DOS.BAS"
    File = Initfilesystem(1)
,Jeśli inicjacja systemu plików przebiegła prawidłowo to sprawdzamy czy na
karcie istnieje plik log.txt a jeśli nie to go tworzymy
    If File = 0 Then
      Strvalue = Dir( „log.txt")
      If Len(strvalue) = 0 Then  ,Tworzymy plik
        Open „log.txt" For Output As #1
        Close #1
      End If
      Else
      Do
        Call Blinking(3)  ,Błąd inicjacji systemu plików
        Waitms 500
      Loop
    End If
  End If
  Else
  Do
    Call Blinking(1)  ,Brak karty SD w gnieździe
    Waitms 500
  Loop
End If
```

Na CD: karty katalogowe i noty aplikacyjne elementów oznaczonych w wykazie elementów kolorem czerwonym

włączeniu zasilania, rejestracja danych jest nieaktywna. Należy podkreślić, że przed każdym wyłączeniem zasilania wspomniana rejestracja **musi być zatrzymana** by nie doszło do sytuacji, gdy w trakcie zapisu danych na kartę dojdzie do odłączenia zasilania układu, co może spowodować uszkodzenie plików.

Dla ograniczenia poboru mocy program obsługi wyłącza komparator analogowy i Watchdog. Za pomocą fuse bitów są wyłączane Brown-out detector i On-chip Debug System. Na **rysunku 4** pokazano diagram funkcjonalny obrazujący konstrukcję programu obsługi urządzenia oraz zastosowane mechanizmy programowe.

Uważny Czytelnik zauważy z pewnością, iż korzystając z przewidzianej, programowej możliwości sterowania zasilaniem karty SD (tranzystor T1 i rezystor R3) można byłoby jeszcze bardziej ograniczyć zużycie prądu podczas uśpienia mikrokontrolera poprzez wyłączanie zasilania tejże karty w tym stanie pracy urządzenia (tak jak ma to miejsce podczas problemów z inicjalizacją jej sterownika). Z jednej strony jest to bardzo dobry pomysł z drugiej zaś tego typu rozwiązanie wydłużyłoby znacznie czas obsługi przerwania INT0 (a więc czas wybudzenia), gdyż po włączeniu zasilania karty należałoby każdorazowo inicjalizować jej sterownik, a trzeba wiedzieć, iż właśnie podczas inicjalizacji sterownika układy interfejsowe karty pobierają największy prąd. Biorąc to wszystko pod uwagę, zrezygnowano z rozwiązania tego typu pozostawiając zasilanie karty SD włączone na stałe. Inną sprawą jest, iż każdorazowa inicjalizacja sterownika karty SD zabierając cenny czas procedury INT0 nie pozwoliłaby na tak dużą częstość pomiarów przyspieszenia.

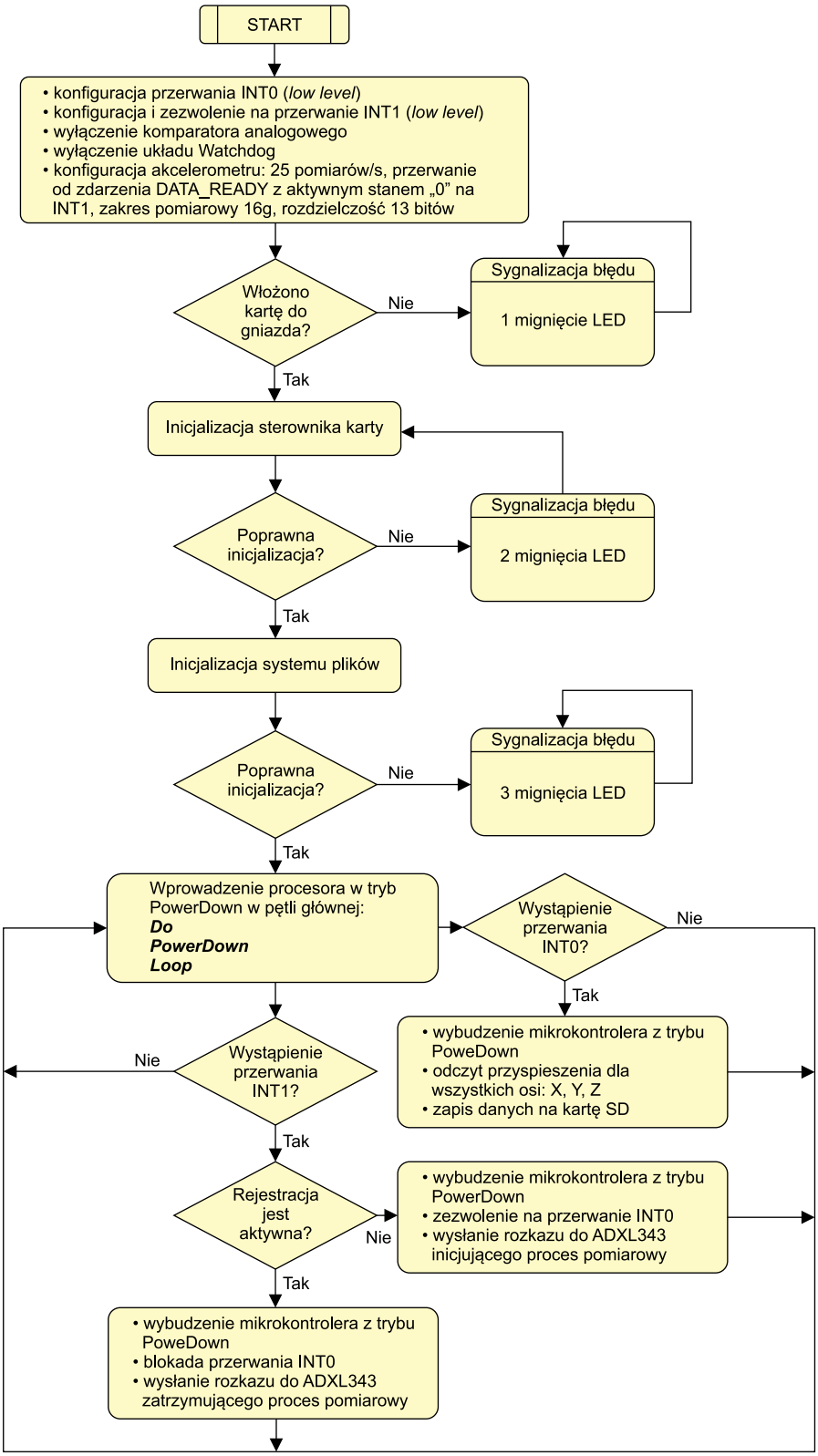
Montaż

Schemat montażowy rejestratora pokazano na **rysunku 5**. Rejestrator ma zwartą, niewielką konstrukcję wykonaną z użyciem elementów SMD. Są one montowane na obu warstwach płytki. Z uwagi na dość kłopotliwą w montażu obudowę akcelerometru (typu LGA-14), trzeba zastosować stację lutowniczą na gorące powietrze oraz odpowiednie topniki.

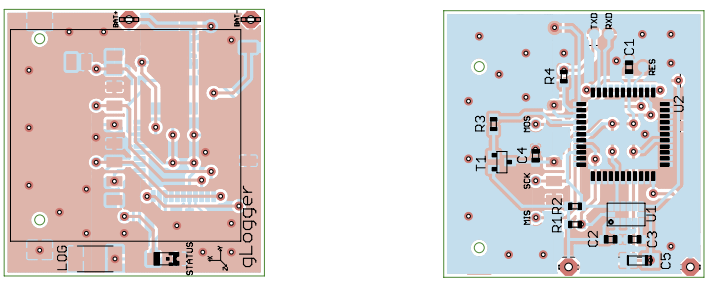
Montaż rozpoczynamy od wlutowania wszystkich elementów biernych, następnie półprzewodnikowych. Na samym końcu mocujemy elementy mechaniczne. Należy uważać, by nie uszkodzić termicznie komponentów, zwłaszcza półprzewodnikowych.

Poprawnie zmontowany układ powinien działać po włączeniu zasilania. Należy zwrócić szczególną uwagę na polaryzację źródła zasilania, ponieważ urządzenie nie zostało zabezpieczone przed przypadkowym odwróceniem polaryzacji.

Robert Wołgajew, EP



Rysunek 4. Diagram funkcjonalny programu obsługi gLoggera



Rysunek 5. Schemat montażowy gLoggera