

pomocą multimetru, który ma tę funkcjonalność, a jednocześnie nie jest to czymś niezwykłym. W następnym kroku dodajemy w rejestrze **0x08** na pozycjach bitów o numerach **3:0** wewnętrzne kondensatory, które mamy do dyspozycji w układzie scalonym. Są to pojemności z zakresu od 0 do 120 pF, z krokiem co 8 pF. Udało mi się osiągnąć wartość częstotliwości na pinie **IRQ = 31,28 kHz**, a więc łatwo obliczyć, że częstotliwość rezonansowa anteny to 500,48 kHz ($31,28 \times 16 \rightarrow 500,48$). W moim wypadku wystarczyło dodanie trzech kondensatorów, czyli zapisanie do rejestru 0x08 komendy 0x03. Oczywiście, może to być inna wartość z przedziału 0x01 do 0x0F w zależności od układu i wartości elementów RC w obwodzie anteny. Wartość dodanych kondensatorów zapamiętujemy – będzie nam ona potrzebna przy inicjalizacji układu.

teraz przechodzimy do inicjalizacji układu. Według noty, na początku przywracamy ustawienia fabryczne, dodajemy kondensatory do obwodu anteny, kalibrujemy oscylatory, a następnie włączamy i wyłączamy generator **TRCO**. Dalsze ustawienia to już konfiguracja rejestrów czułości układu, progu zadziałania watchdoga, progu szumów i tym podobne. Gotową funkcję inicjalizacji, napisaną w języku C, którą zastosowałem w swojej aplikacji zamieszczono na **listingu 1**.

Oczywiście, można modyfikować zawartość rejestrów umieszczonych pod adresami 0x00 i 0x01, a także dodać ustawienia w rejestrze **0x02** (bity **3:0** – rej. SREJ, wykres w nocie aplikacyjnej). Najważniejsze jest pięć pierwszych pozycji z funkcji – resztę można dowolnie modyfikować w celu uzyskania najlepszego efektu końcowego. Oczywiście, bezwzględnie należy sprawdzić w karcie katalogowej do czego służą poszczególne bity i nie wolno modyfikować ich na zasadzie chybił-trafił. Nie należy też przesadzać z ustawieniem zbyt wysokiego poziomu progu szumów i watchdoga, ponieważ ma to bezpośredni wpływ na odległość do wykrywanych wyładowań. A na koniec jeszcze jedna przestroga – jeżeli płytka odbiornika będzie umieszczona blisko komputera lub telefonu komórkowego, to układ będzie zgłaszał przerwanie od szumów i zakłóceń.

Układ AS3935 po wykryciu pioruna generuje przerwanie, na które musimy zareagować. W swojej aplikacji, w procedurze obsługi przerwania ustawiam bit sygnalizujący odebrane wyładowanie, a w pętli głównej programu sprawdzam jego stan, jak na **listingu 2**.

Za pomocą terminala np. Putty możemy podglądać zdarzenia. Ważne, aby po wejściu w funkcję odczytującą dane opóźnić działanie programu przed odczytem rejestru **0x03** (bity **3:0**), czyli powodu przerwania (1 – szum, 4 – zakłócenie, 8 – wyładowanie atmosferyczne). W aplikacji reagujemy tylko na wartość „8” i odczytujemy rejestr **0x07**, czyli odległość od wyładowania, a następnie wysyłamy do terminala. Jeżeli wszystko działa poprawnie, to możemy w funkcji pominąć wysyłanie danych na terminal i napisać ją inaczej, na przykład tak, jak pokazano na **listingu 3**. Dane o odległości od wyładowania są wysyłane bo głównego procesora stacji pogodowej, w którym są parsowane i prezentowane na wyświetlaczu TFT.

Jeszcze jeden bardzo ważny aspekt. Układ po wykryciu wyładowania np. w odległości 10 kilometrów zapamiętuje tę wartość i jeżeli następne wyładowanie nastąpi w odległości większej, to niestety ponownie zostanie zgłoszone wcześniejsze 10 kilometrów. Jest to trochę uciążliwe, ponieważ jeżeli wyładowanie nastąpi naprawdę blisko, to stale jest pokazywana ta wartość aż do przepełnienia się rejestru, pomimo odsuwania się burzy. Można temu zaradzić manipulując bitem 6 w rejestrze **0x02**. Na pozycję tego bitu wpisujemy kolejno „1” – „0” – „1” i dodajemy to do funkcji **Storm**. Od tej pory mamy zawsze aktualny odczyt odległości.

Układ AS3935 ma też rejestry, z których możemy odczytać energię, jaką wytworzył uderzający piorun. Są one umieszczone pod adresem **0x04**, **0x05** i **0x06**. Niestety, producent nie podaje, w jakich

Listing 1. Inicjalizowanie układu AS3935

```
void AS3935_init(void)
{
    uint8_t cd = 0x96;
    uint8_t cap = 0x03;
    uint8_t a;

    TWI_write_buf(AS3935_ADRES, 0x3C, 1, &cd); // ustawienia fabryczne
    _delay_ms(2);
    TWI_write_buf(AS3935_ADRES, 0x08, 1, &cap); // dodanie kondensatorów wewnętrznych
    _delay_ms(2);
    TWI_write_buf(AS3935_ADRES, 0x3D, 1, &cd); // kalibracja oscylatorów
    a = 0x23;
    TWI_write_buf(AS3935_ADRES, 0x08, 1, &a); // włączenie generatora TRCO
    _delay_ms(2);
    TWI_write_buf(AS3935_ADRES, 0x08, 1, &cap); // wyłączenie generatora TRCO
    a = 0x24;
    TWI_write_buf(AS3935_ADRES, 0x00, 1, &a); // ustawienie wzmocnienia AFE
    a = 0x24;
    TWI_write_buf(AS3935_ADRES, 0x01, 1, &a); // ustawienie WIDTH
}
```

Listing 2. Procedura obsługi przerwania, funkcja wykrywania burzy i pętla główna

```
ISR(INT2_vect)
{
    FLAG = 1;
    LED_ON;
}

void Storm(void)
{
    _delay_ms(5);
    uint8_t buf, buf1;

    TWI_read_buf(AS3935_ADRES, 0x03, 1, &buf);
    uart_puts("PRZERWANIE");
    uart_puts("\r\n");
    uart_puts("AS3935 rejestr INT = ");
    uart_putint(buf, 16);
    uart_puts("\r\n");
    if(buf == 8)
    {
        TWI_read_buf(AS3935_ADRES, 0x07, 1, &buf1);
        uart_puts("AS3935 rejestr DISTANCE = ");
        uart_putint(buf1, 10);
        uart_puts("km");
        uart_puts("\r\n");
    }
    FLAG = 0;
    LED_OFF;
}

while(1)
{
    if(FLAG)Storm();
}
```

Listing 3. Modyfikacja funkcji wykrywającej burzę

```
void Storm(void)
{
    _delay_ms(5);
    uint8_t buf, buf1;

    TWI_read_buf(AS3935_ADRES, 0x03, 1, &buf);
    if(buf == 8)
    {
        TWI_read_buf(AS3935_ADRES, 0x07, 1, &buf1);
        storm = buf1;
        sprintf(bufor, ("AT+storm %d"), storm);
        uart_puts(bufor);
        uart_puts("\r\n");
    }
    FLAG = 0;
}
```

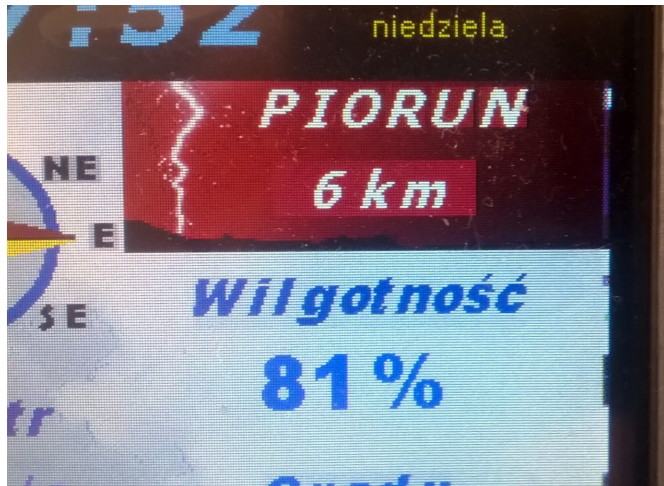
jednostkach jest kalkulowana ta energia i jak jest szacowana, więc myślę, że należy to potraktować jako ciekawostkę.

Generalnie, układ pracuje poprawnie od ponad 6 miesięcy w mojej stacji pogodowej. Zdarzało się, że sygnalizował wyładowanie, którego nawet nie było słyhać. Największa do tej pory odległość, z jakiej wykrył piorun pracując na zewnątrz budynku to 33 kilometry.

Układ AS3935 można kupić poprzez stronę internetową firmy Austria Microsystems. Dostępne są również gotowe moduły ze wszystkimi elementami pod nazwą MOD-1016. Jeżeli podejmiemy decyzję o zakupie samego układu i zaprojektowanie własnej płytki, to trzeba jeszcze uwzględnić zakup cewki MA5532-AE firmy Coilcraft przeznaczonej do użycia z układem AS3935 w obwodzie antenowym. Trzeba również przewidzieć miejsce na dwa kondensatory w obwodzie anteny. Producent w nocie podaje wartość 1 nF, ale warto mieć alternatywę w postaci dwóch kondensatorów np. 680 pF + 270 pF, ponieważ może się okazać, że 1 nF to za dużo (szczególnie przy 10% tolerancji pojemności) i nie uda nam się precyzyjnie dostroić anteny.



Fotografia 3. Wyniki pomiaru prezentowane w oknie programu terminala



Fotografia 4. Wyniki pomiaru pokazywane przez stację pogodową

Listing 4. Przykładowy plik nagłówkowy biblioteki obsługi AS3935

```
#ifndef AS3935_H_
#define AS3935_H_
#define AS3935_ADRES 0x06
volatile uint8_t FLAG;
uint8_t storm;
void AS3935_init(void);
void Storm(void);
#endif /* AS3935_H_ */
```

Na fotografii 3 pokazano wyniki pomiarów przesyłane do terminala, natomiast na fotografii 4 sposób ich prezentacji na wyświetlaczu LCD stacji pogodowej.

Zachęcam do wykonania w programie oddzielnych plików bibliotecznych AS3935.c i AS3935.h mimo tego, że cała „biblioteka”

to zaledwie dwie funkcje. Przykładowy plik nagłówkowy (.h) pokazano na listingu 4. Biblioteki do obsługi UART oraz I²C wykonano na podstawie książek Mirosława Kardasia „Mikrokontrolery AVR język C – podstawy programowania” oraz „Język C pasja programowania mikrokontrolerów 8-bitowych”. Nota aplikacyjna jest dostępna w Internecie, w tym na stronie producenta – firmy Austria Microsystems.

Powodzenia w pracy z układem i czekamy na wiosenną burzę!

Marek Rębecki

Wygraj płytkę deweloperską Microchip SAMA5D27-SOM1-EK1



Wartość nagrody to ponad 900 zł!

Firma Microchip organizuje konkurs dla czytelników Elektroniki Praktycznej, w ramach którego mogą oni wygrać płytkę Microchip SAMA5D27-SOM1-EK1 (model ATSAMA5D27-SOM1-EK1).

Płytkę deweloperską SAMA5D27-SOM1-EK1 to platforma do szybkiego prototypowania i testowania projektów z modułem SAMA5D27-SOM1, który ułatwia tworzenie przemysłowych systemów mikroprocesorowych.

Moduł zawiera układ ATSAMA5D27C-D1G-CU z rdzeniem ARM Cortex-A5 i znacząco upraszcza projektowanie, gdyż ma zintegrowane funkcje zarządzania mocą, pamięć nieulotną, interfejs ethernetowy i szybką pamięć DDR2, mając przy tym postać niedużej, jednostronnej płytki drukowanej. Na płytce znalazło się 128 MB pamięci DDR2 DRAM.

Płytkę ma wiele obwodów peryferyjnych oraz pozwala na rozszerzenie swoich funkcji m.in. poprzez użycie dwóch złączy mikroBUS, obsługujących płytki MikroElektronika click.

Aby wygrać płytkę Microchip SAMA5D27-SOM1-EK1 wystarczy zarejestrować się na stronie <http://bit.ly/2zHVSxS>.