

Detektor wyładowań atmosferycznych

Wyładowania atmosferyczne – błyskawice – towarzyszą ludziom od zawsze. To dosyć częste zjawisko kiedyś kojarzone było z siłami nadnaturalnymi czy boskimi, teraz jest już doskonale zrozumiałe, jednakże nadal fascynuje. Nowoczesna technologia pozwala na wykrywanie wyładowań atmosferycznych nawet z dużej odległości. Meteorolodzy wykorzystują dane na temat wyładowań do przewidywania i monitorowania pogody. Teraz, dzięki elektronice, każdy może samodzielnie zbudować urządzenie do detekcji wyładowań atmosferycznych.

Systemy wykrywania aktywności wyładowań atmosferycznych wykorzystują relatywnie nieskomplikowane systemy radiowe. Podczas uderzenia błyskawica emituje promieniowanie elektromagnetyczne w pasmie radiowym. Dzięki zastosowaniu czułego odbiornika RF i przetwarzaniu odebranego sygnału za pomocą złożonego algorytmu system może nie tylko wykryć fakt, że gdzieś uderzyła błyskawica, ale także określić, jak daleko miało to miejsce.

Układ scalony, taki jak AS3935, może precyzyjnie określić, czy wykryta aktywność elektryczna jest spowodowana piorunem. Układ ten precyzyjnie szacuje prawdopodobną odległość na podstawie intensywności sygnału. Za odległość przyjmowana jest odległość od początku burzy, gdzie zaczyna się tworzyć błyskawica, a nie od punktu, gdzie uderzyła. Chip może określić odległość z wystarczającym przybliżeniem między maksymalnie 40 km a minimum 1 km. Algorytm używany w systemie pozwala na odróżnienie, z pewnym przybliżeniem, błyskawic (wyładowań atmosferycznych) od wyładowań elektrycznych wytwarzanych na przykład przez cewki zapłonowe i świece zapłonowe, silniki elektryczne szczotkowe czy kuchenki mikrofalowe.

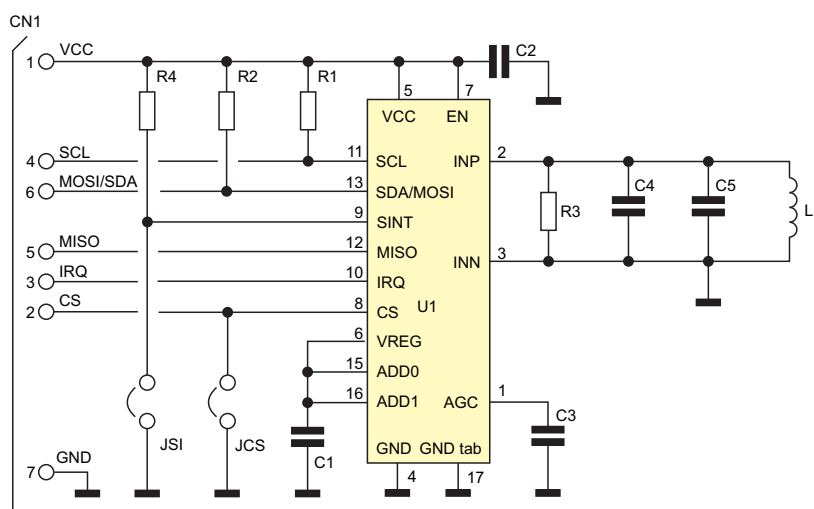
Scalony detektor wyładowań atmosferycznych

W poniższym projekcie wykorzystamy scalony detektor wyładowań atmosferycznych



– AS3935 – oraz moduł Arduino z wgranym odpowiednim oprogramowaniem. W pierwszej kolejności podłączyć musimy scalony detektor do płytki Arduino. Najprościej jest zrobić to z wykorzystaniem prostego modułu,

na którym znajdzie się układ oraz goldpiny. Posłużą one do połączenia go z wyprowadzonymi Arduino. Na module znajdują się też potrzebne do działania elementy dyskretnie czy kondensator filtrujący zasilanie. Schemat



Rysunek 1. Schemat ideowy modułu do detekcji wyładowań atmosferycznych

takiego modułu pokazany jest na **rysunku 1**. Jeśli nie chcemy konstruować go samodzielnie, tego rodzaju moduły dostępne są również w handlu.

Możemy wyróżnić dwa podstawowe zespoły elementów, towarzyszące AS3935 – tor analogowy, złożony z układu RLC (L1, R3, C4 i C5) oraz cyfrowy interfejs, zawierający rezystory podciągające, konieczne do podłączenia układu do mikrokontrolera.

Układ zawiera wbudowany stabilizator napięcia. Dzięki temu AS3935 może być zasilany przez wewnętrzny stabilizator napięcia lub bezpośrednio przez zewnętrzne zasilanie. Użycie wewnętrznego regulatora napięcia zwiększy pobór prądu o około 5 μ A. Aby włączyć wewnętrzny regulator napięcia, piny VDD i EN_VREG muszą zostać podłączone do napięcia zasilania. Pojemność większa niż 1 μ F musi być podłączona od pinu VREG do masy, aby zapewnić stabilne działanie stabilizatora napięcia. Nominalnie napięcie wyjściowe z wbudowanego stabilizatora wynosi 3 V. Aby zasilić AS3935 bezpośrednio zewnętrznym źródłem zasilania (np. z zewnętrznego stabilizatora w układzie), pin EN_VREG musi zostać podłączony do masy. Zarówno VDD, jak i VREG muszą być wtedy podłączone do napięcia zasilania.

Interfejs cyfrowy

Omawiany układ – detektor wyładowań atmosferycznych – wyposażony jest w interfejs szeregowy do komunikacji z mikrokontrolerem. Układ może być skonfigurowany do pracy z wykorzystaniem interfejsu I²C lub SPI. W opisywanym projekcie zastosowano ten drugi z interfejsów. Wykorzystuje on cztery linie cyfrowe:

- MISO (*Master In, Slave Out*) – linia wykorzystywana przez układ do wysyłania danych do kontrolera, w naszym przypadku do modułu Arduino
- MOSI (*Master out, Slave In*) – linia, którą moduł Arduino będzie wykorzystywał do przesyłania danych do układu
- CS (*Chip Select*) – linia, z pomocą której master (moduł Arduino) kontroluje pracę interfejsu szeregowego
- SCL (*Serial Clock*) – zegar transmisji szeregowy, wykorzystywany do synchronizacji transmisji w liniach MISO oraz MOSI.

Do wyboru pomiędzy oboma interfejsami szeregowymi służy nóżka numer 9 układu, oznaczona SINT. Podanie stanu wysokiego na ten pin powoduje, że układ działać będzie w trybie I²C, a po podaniu stanu niskiego wykorzystywać będzie interfejs SPI. W module, którego schemat pokazano na rysunku 1, stan tego pinu kontrolowany jest przez zworę JS1. Aby wybrać interfejs SPI, zworka ta musi być zwarta.

Sekcja analogowa

Antena zewnętrzna jest bezpośrednio podłączona do frontendu analogowego (AFE),

który wzmacnia i demoduluje odbierany sygnał. Watchdog stale monitoruje wyjście AFE i ostrzega zintegrowany blok detekcji wyładowania atmosferycznego w przypadku sygnału przychodzącego.

Moduł AFE wzmacnia i demoduluje sygnał AC odbierany przez antenę systemu. Układ AS3935 wykorzystuje technikę odbioru wąskopasmowego sygnału o częstotliwości środkowej 500 kHz i szerokości około 33 kHz do detekcji wyładowań atmosferycznych. Wzmocnienie AFE jest w zasadzie stałe w całym paśmie pracy anteny. Osiągnięto to poprzez zapewnienie szerszego pasma AFE niż pasmo anteny. Wzmocnienie AFE zostało zoptymalizowane dla dwóch środowisk pracy – w pomieszczeniu i na dworze. Domyślnie system ustawiony jest na pracę w pomieszczeniu, ale można zmienić to z pomocą pojedynczego rejestru konfiguracyjnego.

Strojenie anteny dla układu AS3935 jest dosyć proste – składa się ona z podłączonego do systemu równoległego rezonatora LC z dodanym opornikiem. Wymagania co do tego układu są bardzo proste – ma on charakteryzować się częstotliwością rezonansową $f_r \approx 500$ kHz i dobrocią $Q \geq 15$. Częstotliwość rezonansowa takiego układu LC opisana jest równaniem 1, a dobroć równaniem 2.

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad (1)$$

$$Q = R \cdot \sqrt{\frac{C}{L}} \quad (2)$$

W układzie zastosowano cewkę MA5532-AE firmy Coilcraft, która została opracowana we współpracy z producentem układu AS3935. Cewka ta charakteryzuje się indukcyjnością równą 100 μ H i rezystancją stałoprądową uzwojenia równą 6 Ω . Jeśli teraz uzupełnimy ten układ kondensatorem o pojemności 1 nF, to częstotliwość rezonansowa $f_r \approx 503,3$ kHz. Jeśli teraz do układu tego dołożymy równolegle rezystor o oporności 10 k Ω uzyskamy dobroć $Q \approx 31,6$. Parametry te w pełni satysfakcjonują wymagania układu.

Ustawiając rejestry cyfrowe w układzie, można dodawać i odejmować wewnętrzne pojemności do układu LC, aby odstroić częstotliwość rezonansową o $\pm 3,5\%$. Układ może też mierzyć częstotliwość rezonansową, więc proces strojenia anteny przeprowadzać można w pełni automatycznie.

Sygnał wyjściowy z AFE jest monitorowany przez watchdog. W przypadku, gdy sygnał przekroczy pewien próg szerokości – czasu trwania – układ przechodzi w tryb weryfikacji sygnału. Poziom tego progu można ustawić w jednym z rejestrów układu. Po zwiększeniu progu, AS3935 może być bardziej odporny na zakłócenia. Jednak spowoduje to również, że czujnik będzie mniej czuły na słabsze sygnały z odległych wyładowań.

Sygnał wyjściowy AFE jest również wykorzystywany do estymacji poziomu szumów – poziomu odniesienia. Poziom szumu jest stale porównywany z napięciem odniesienia (próg szumu). Ilekroć poziom szumu przekroczy ustalony próg, układ AS3935 generuje przerwanie (INT_NH), aby poinformować jednostkę zewnętrzną (np. mikrokontroler), że nie może działać poprawnie z powodu wysokiego szumu wejściowego odbieranego przez antenę. Możliwa jest konfiguracja poziomu szumu granicznego dla urządzenia.

Algorytm działania układu AS3935

Blok detekcji systemu sprawdza sygnał względem predefiniowanego wzorca. Układ jest w stanie rozróżniać sygnały spowodowane uderzeniami pioruna i sygnały generowane przez sztuczne źródła – zakłócenia. W przypadku, gdy sygnał jest zaklasyfikowany jako zakłócenie, zdarzenie jest odrzucane, a czujnik automatycznie powraca do trybu nasłuchu. Gdy zdarzenie sklasyfikowane jest jako uderzenie pioruna, blok oceny statystycznej dokonuje oszacowania odległości do czoła burzy.

Algorytm detekcji wyładowań wykorzystuje trzy kroki:

1. Walidacja sygnału – decyzja, czy wykryty sygnał jest w istocie wyładowaniem atmosferycznym.
2. Obliczenie energii uderzenia – wyznaczenie całkowitej energii pojedynczego zdarzenia
3. Statystyczna estymacja odległości od wyładowania – ocena odległości uderzenia na podstawie danych statystycznych i obliczonej energii uderzenia.

W przypadku, gdy odebrany sygnał nie ma charakterystyki kształtu związanego z wyładowaniem atmosferycznym, walidacja sygnału kończy się automatycznie niepowodzeniem i zdarzenie jest klasyfikowane jako inne zakłócenie. W takim przypadku obliczenia energii i statystyczne szacowanie odległości nie są wykonywane, a czujnik automatycznie powraca do trybu nasłuchu.

Najkrótszy czas między dwoma uderzeniami pioruna, które AS3935 może wykryć i zanalizować, wynosi około jednej sekundy. Gdy sygnał zostanie zaklasyfikowany jako zakłócenie, czujnik jest dezaktywowany na dalsze 1,5 sekundy. Ponieważ czas trwania sygnałów zakłócających może się zmieniać, czas przestoju czujnika uniemożliwia wielokrotne wyzwalenie czujnika z powodu dłuższych zakłóceń.

Podczas fazy walidacji sygnału analizowany jest jego kształt i przebieg. Czujnik rozróżnia sygnały naturalne i sztuczne zakłócenia. Algorytm dyskryminacji impulsów może być konfigurowany za pomocą rejestrów konfiguracyjnych w systemie. Zmiany w tym zakresie pozwalają uczynić system

Listing 1.

```

#include <SPI.h>
#include <AS3935.h>
#include <SoftwareSerial.h>
#define LCD_pin 5 // pin do którego podłączony jest wyświetlacz
#define CS_pin 3 // pin do którego podłączona jest linia CS detektora
#define IRQ_pin 2 // pin do którego podłączone jest przerwanie z detektora

const int NoDetect=30;
int counter;
int NumDisturber=0;

SoftwareSerial LCD = SoftwareSerial(0, LCD_pin);

void printAS3935Registers();

// Prototyp funkcji zapewniającej komunikację poprzez SPI z układem
// AS3935. Zdefiniowana jest poniżej. Funkcja wymagana jest przez bibliotekę
// układu do komunikacji. To użytkownik musi ją zaimplementować, dzięki czemu
// biblioteka dla AS3935 jest w dużej mierze agnostyczna dla sprzętu.
// Biblioteka wymaga funkcji o dokładnej takiej sygnaturze, nie może to być
// metoda żadnej klasy w C++, jak jest z większością funkcji w bibliotekach Arduino.
// Upewnij się, że funkcja ta nie używa pinu CS - biblioteka zajmuje się nim sama.

byte SPITransfer(byte sendByte);

int tunecap; // zmienna dla wartości rejestru kalibracyjnego

// Obsługa przerwania dla AS3935. Zmienne definiowane w przerwaniu musza
// posiadać słowo kluczowe volatile, inaczej program może nie skompilować
// się poprawnie z uwagi na optymalizacje kodu.
void AS3935Irq();

// Pierwszy parametr - funkcja do transferu przez SPI transfer function,
// Drugi parametr - pin Arduino, wykorzystany jako linia CS
// Trzeci parametr - pin Arduino wykorzystany jako linia IRQ.
// Dobrze jest wykorzystać do przerwania pin, który predefiniowano do takiego zastosowania,
// dzięki temu możliwe jest wykorzystanie wbudowanej funkcji attachInterrupt
// w szkicu, do wykrywania przerwania. Biblioteka wewnętrznie sprawdza wartość tego pinu
// podczas kalibracji.
AS3935 AS3935(SPITransfer, CS_pin, IRQ_pin);

void setup(){
    serLCDInit();
    backlightOn();
    clearLCD();
    lcdPosition(0,3);
    LCD.print(„Lightning”);
    lcdPosition(1,5);
    LCD.print(„Sensor”);
    delay(1500);
    clearLCD();

    Serial.begin(9600);
    // Najpierw uruchamiamy SPI, potem ustawiamy parametry
    SPI.begin();
    // układ korzysta z trybu SPI MODE1
    SPI.setDataMode(SPI_MODE1);
    // Maksymalny zegar SPI dla tego układu 2 MHz,
    // ale nigdy nie ustawiamy zegara na 500 kHz
    // gdyż na tej częstotliwości działa detektor
    // wyładowań i prowadzić może to do interferencji.
    SPI.setClockDivider(SPI_CLOCK_DIV16);
    // Układ wysyła najpierw bit MSB.
    SPI.setBitOrder(MSBFIRST);
    // Zresetujemy wszystkie wewnętrzne rejestry do stanu domyślnego.
    AS3935.reset();
    delay(10);
    AS3935.setOutdoors();
    // Zapisujemy wartość 2 w rejestrze odpowiedzialnym za poziom szumów
    AS3935.registerWrite(AS3935_NF_LEV,2);
    // i uruchamiamy kalibrację.
    // Jeśli układ nie będzie mógł zestroić obwodu odbiorczego z wymagana tolerancja, to
    // funkcja kalibracyjna zwróci wartość fałszywa. Inaczej zwróci true.
    if(!AS3935.calibrate())
        Serial.println(„Strojenie się nie powiodło. Sprawdź okablowanie i sensor.”);
    // Teraz możemy pobrać i wyświetlić wartość rejestru kalibracyjnego TUN_CAP
    // który będzie w zakresie od 0 do 15.
    tunecap=AS3935.registerRead(AS3935_TUN_CAP); // Wewnętrzna kalibracja.
    Serial.print(„Rejestr kondensatora strojącego wynosi „);
    Serial.println(tunecap);
    // Możemy także poinformować AS3935, że znajdujemy się na dworzu. Odkomentujmy kolejną linię
    // AS3935.setOutdoors();
    // Dalej musimy uruchomić wykrywanie wyładowań atmosferycznych.
    AS3935.enableDisturbers();
    // raz wypisać wartość rejestrów konfiguracyjnych.
    printAS3935Registers();
    attachInterrupt(0,AS3935Irq,RISING);
}

void loop(){
    // W głównej petli sprawdzamy, czy przerwabue zostało wyzwolone. Jest to trochę
    // kontrproduktywne - nie tak powinny działać przerwania, ale tak jest zrealizowane to w Arduino.
    // Normalnie można by oprzeć się na sprzętowym przerwaniu, a cały układ uspic na czas
    // oczekiwania na przerwanie, aby zredukować pobór mocy.

    // Większość kodu obsługi znajduje się w funkcji obsługującej przerwanie,
    // w głównej petli programu piszemy jedynie o oczekiwaniu na wyzwolenie,
    // aby upewnić się, że program działa cząly czas poprawnie.
    delay(1000);
    Serial.println(„Oczekiwanie...”);
    if (counter==0){
        NumDisturber=0;
        counter=NoDetect;
        clearLCD();
        lcdPosition(0,1);
        LCD.print(„Brak wyładowania”);
        lcdPosition(1,4);
        LCD.print(„wykryto”);
    } else {

```

mniej podatnym na zakłócenia kosztem czułości – system ustawiony jako bardziej odporny na zakłócenia będzie miał większe problemy z wykrywaniem wylądowań znajdujących się daleko.

Jeśli odebrany sygnał zostanie sklasyfikowany jako błyskawica, obliczana jest jego energia. Wynik obliczenia energii jest następnie zapisywany w rejestrach układu.

Wartość ta jest tylko bezwymiarową liczbą i nie ma fizycznego znaczenia – to tylko estymator wyznaczony na podstawie odebranego sygnału. Jest on potrzebny do oszacowania odległości od wykrytego wylądowania.

Finalnie AS3935 generuje szacunkową odległość do czoła wykrytej burzy. Ocena ta odbywa się na podstawie analizy statystycznej. Blok szacowania odległości to miejsce,

w którym analiza ta jest prowadzona. Blok ten estymuje odległość od czoła burzy na podstawie danych statystycznych, zebranych z pewnej liczby poprzednich wykrytych zdarzeń. Algorytm automatycznie usuwa stare dane z pamięci. Szacowana odległość jest zapisywana w rejestrze układu i odpowiada odległości od czoła burzy w kilometrach. Sam algorytm szacowania odległości od burzy

Listing 1. cd.

```

        counter=counter - 1;
    }
}

void printAS3935Registers(){
    int noiseFloor = AS3935.getNoiseFloor();
    int spikeRejection = AS3935.getSpikeRejection();
    int watchdogThreshold = AS3935.getWatchdogThreshold();
    Serial.print(„Poziom szumu wynosi: „);
    Serial.println(noiseFloor,DEC);
    Serial.print(„Poziom odrzucania szpilek wynosi: „);
    Serial.println(spikeRejection,DEC);
    Serial.print(„Próg watchdoga wynosi: „);
    Serial.println(watchdogThreshold,DEC);
}

// To pełna implementacja funkcji do transferu poprzez SPI, która podawana jest do
// biblioteki AS3935. Można to zmienić na własną implementację komunikacji.
byte SPITransfer(byte sendByte){
    return SPI.transfer(sendByte);
}

// To jest funkcja obsługująca przerwanie z AS3935. Nic nie zwraca i nie przyjmuje
// żadnych argumentów. Kod tutaj powinien być krótki i wykonywać się szybko.
void AS3935Irq(){
    // Cały kod, odpowiedzialny za wyświetlanie informacji na LCD przesunięty został do przerwania.
    // Pierwszym krokiem jest sprawdzenie co jest źródłem przerwania, jak tylko informacja o nim
    // dojdzie. Po odczytaniu rejestru przerwania pin IRQ przejdzie w stan niski.
    int irqSource = AS3935.interruptSource();
    // Wartość ta definiuje co jest źródłem przerwania:
    // 0 - zbyt wysoki poziom szumu
    // 2 - wykryto zakłócenie
    // 3 - wykryto wylądowanie atmosferyczne.
    if (irqSource & 0b0001) Serial.println(„Poziom szumu zbyt wysoki, spróbuj zmienić ustawienie poziomu szumu.”);

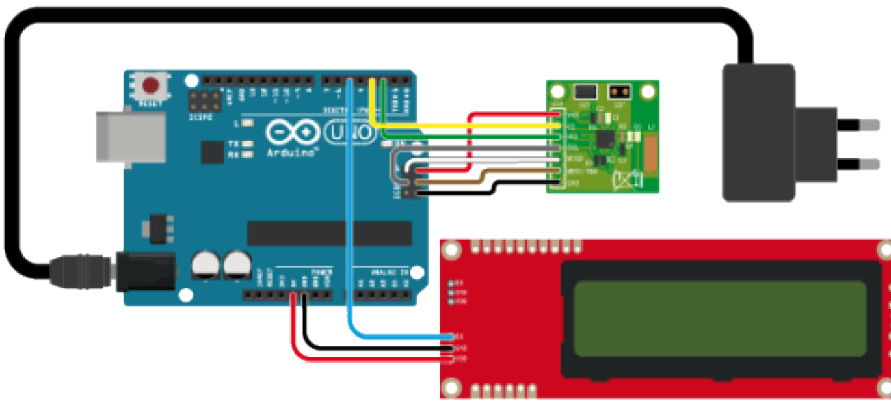
    if (irqSource & 0b0100){
        NumDisturber+=1;
        Serial.println(„Wykryto zakłócenie”);
        clearLCD();
        lcdPosition(0,0);
        LCD.print(„Liczba wylądowań: „);
        LCD.print(NumDisturber,DEC);
        counter=NoDetect;
    }

    if (irqSource & 0b1000){
        // W tym przypadku uzyskać musimy nie tylko informacje o tym, że wykryto wylądowanie, a także
        // ile ich wykryto, ale także o odległości od wylądowania w kilometrach, gdzie 1 to burza
        // poniżej minimalnej odległości detekcji, a 63 to wylądowanie powyżej granicy. Wszystkie
        // pozostałe wartości, to odległość w kilometrach.
        int strokeDistance = AS3935.lightningDistanceKm();
        if (strokeDistance == 1){
            Serial.println(„Burza znajduje się nad Tobą, uważaj!”);
            lcdPosition(1,1);
            Serial.println(„Burza nad Tobą”);
            lcdPosition(1,3);
            Serial.println(„UWAŻAJ!”);
            counter=NoDetect;
        }

        if (strokeDistance == 63){
            Serial.println(„Wykryto wylądowanie poza zasięgiem detekcji.”);
            lcdPosition(0,2);
            Serial.println(„Poza zasięgiem”);
            lcdPosition(1,0);
            Serial.println(„wykryto wylądowanie”);
            counter=NoDetect;
        }

        if (strokeDistance < 63 && strokeDistance > 1){
            Serial.print(„Wykryto wylądowanie w odległości „);
            Serial.print(strokeDistance,DEC);
            Serial.println(„ kilometrów.”);
            lcdPosition(1,0);
            LCD.print(„Odległość: „);
            LCD.print(strokeDistance,DEC);
            LCD.print(„km”);
            counter=NoDetect;
        }
    }
}
}

```



Rysunek 2. Schematyczne przedstawienie podłączenia modułu z detektorem wyładowań atmosferycznych, wyświetlacza LCD i modułu Arduino

Tabela 1. Spis połączeń pomiędzy modułem Arduino a płytką detektora wyładowań atmosferycznych

Numer pinu w Arduino	Nazwa sygnału	Numer pinu w module detektora
Pin 2 gniazda ICSP 2	VCC – zasilanie	1
D3	CS	2
D2	IRQ	3
Pin 3 gniazda ICSP 2	SCL	4
Pin 1 gniazda ICSP 2	MISO	5
Pin 4 gniazda ICSP 2	MOSI	6
Pin 6 gniazda ICSP 2	GND – masa	7

Tabela 2. Spis połączeń pomiędzy modułem Arduino a płytką wyświetlacza LCD

Numer pinu w Arduino	Nazwa sygnału
5 V	VCC – zasilanie
D5	RX
GND	GND – masa

jest wbudowany w układ i nie jest dostępny z zewnątrz.

Podłączenie do Arduino

Układ AS3935 wymaga jedynie podłączenia kilku linii, aby działać. Do układu podłączyć trzeba oczywiście masę i zasilanie, interfejs SPI, a także dodatkową linię – przerwanie (oznaczone IRQ na rysunku 1). Na tym pinie detektor burzy wystawia stan wysoki w momencie, gdy wykryte zostaje wyładowanie atmosferyczne.

Listę połączeń pomiędzy detektorem (którego schemat pokazano na rysunku 1) i modułem Arduino zawarto w tabeli 2. Dodatkowo, do modułu Arduino, podłączony został wyświetlacz LCD połączony poprzez interfejs szeregowy.

Na rysunku 2 schematycznie zaprezentowano połączenia opisane w tabeli 1

oraz tabeli 2. Pamiętać oczywiście należy, że duża część z opisanych powyżej połączeń nie jest niezmienna – piny takie jak CS czy programowo generowany interfejs szeregowy mogą być podpięte do innego wyprowadzenia mikrokontrolera, jeżeli tylko odpowiednio skonfiguruje się to w kodzie programu.

Oprogramowanie

Zadaniem modułu Arduino Uno jest analiza sygnału dostarczanego przez czujnik, przetworzenie go i wyświetlenie na wyświetlaczu LCD poprzez port szeregowy. System prezentuje szereg informacji na temat wyładowania: liczbę wykrytych wyładowań elektrycznych, które można uznać za jego odgąlenia oraz szacowaną odległość od wyładowania.

Aby sterować wyświetlaczem szeregowym, użyto biblioteki do emulowania portu szeregowego SoftwareSerial.h, która

umożliwia zwolnienie sprzętowego portu UART dla innych aplikacji, które go potrzebują. Płytkę czujnika jest podłączona przez magistralę SPI. Interfejs ten pozwala na odbieranie wszystkich danych z sensora i wysyłanie informacji konfiguracyjnych. Dzięki algorytmom przetwarzania sygnału radiowego, zaimplementowanym w układzie, Arduino nie musi prowadzić już żadnej analizy sygnału.

/CS (sygnał aktywny na niskim poziomie logicznym) jest obsługiwany przez wyjście cyfrowe 3 modułu Arduino. Skonfigurowane jest w szkicu jako wyjście. Wejście przerwania – IRQ – jest natomiast podłączone do cyfrowego wejścia numer 2 Arduino. Linia IRQ powiadamia Arduino o możliwej detekcji wyładowania przez układ. Gdy IRQ wraca do zera logicznego, następuje przerwa co najmniej 2 ms przed ponownym odczytaniem rejestru.

Wyświetlacz LCD jest zasilany z Arduino przez linie 5 V i GND dla zasilania i sterowany z cyfrowego pinu 5, ustawionego jako wyjście. Z pomocą tego pinu system wysyła szeregowe dane sterujące do wyświetlacza RX poprzez interfejs UART.

Na listingu 1 zaprezentowano kod programu. Wykorzystuje on dwie zewnętrzne biblioteki, z których jedną trzeba pobrać i zainstalować w Arduino – AS3935.h, która pobrana może być tutaj: <http://bit.ly/2NVmKnh>.

Zazwyczaj powiadomienie o burzy jest wyświetlane wraz z liczbą wykrytych zdarzeń w czasie krótszym niż 60 sekund. Jeśli w ciągu 60 sekund nie zostaną wykryte żadne wyładowania atmosferyczne, zostanie wyświetlone powiadomienie o tym fakcie. Jeśli w rejestrze układu scalonego dostępna jest odległość od wyładowania, to ta informacja jest również wyświetlana w drugim wierszu, z komunikatem o treści: Liczba wyładowań: X, gdzie X wskazuje liczbę wykrytych wyładowań. Poniżej można zobaczyć: Odległość: N km, gdzie N to szacowana odległość od wyładowania w kilometrach. Jeśli odległość jest niższa niż minimalna odległość wykrywania (1 km), wyświetlacz pokazuje napis „Burza nad Tobą” a w drugiej linii „UWAŻAJ!”. Analogicznie system informować będzie o wykryciu wyładowania poza maksymalnym zasięgiem detekcji (ok. 60 km).

Nikodem Czechowski

1. <http://bit.ly/2L8NOge>

2. <http://bit.ly/2LJsmhx>

REKLAMA

<https://sklep.avt.pl>