

MOBILNY MONITORING ULATNIAJĄCEGO SIĘ GAZU W POJAZDACH Z INSTALACJĄ LPG

W publikacji zaprezentowano wykorzystanie mikrokomputerów SBC i modułu SIM900, jako elementów systemu do mobilnego monitoringu ulatniającego się gazu w pojazdach z instalacją LPG. Skupiono się przede wszystkim na oprogramowaniu mikrokomputera do współpracy z sensorem wykrywającym ulatniający się gaz oraz oprogramowaniu jednostki centralnej do współpracy z modulem transmisji bezprzewodowej 3G, komunikującym się z mobilnym użytkownikiem systemu. Pokazano ideę programowania tego typu urządzeń, które realizują niezliczone funkcje w systemach automatyki przemysłowej jak i w technologiach motoryzacyjnych. Pominięto platformę - środowisko programistyczne mikrokomputera oraz kody bibliotek wbudowanych i implementowanych biorących udział w kompilacji programu. Brakujący materiał można znaleźć w publikacjach [1], [2], [3].

WSTĘP

Mobilny monitoring ulatniającego się gazu w pojazdach z instalacją LPG jest możliwy dzięki programowalnym systemom, zarządzanym zdalnie z wykorzystaniem transmisji bezprzewodowej. System taki składa z programowalnej platformy detekcyjno-pomiarowej, analizującej skład powietrza w niewrażliwych punktach instalacji oraz terminala bezprzewodowego (telefon komórkowy nowej generacji), do którego przesyłane są dane (komunikaty), w wypadku wystąpienia awarii, spowodowanej ulatnianiem się gazu w instalacji.

W pojazdach samochodowych nie ma miejsca na typowe jednostki komputerowe, w związku z tym do detekcji gazu i jego raportowania wykorzystano w prototypie jednopłytkowy mikrokomputer Arduino o bardzo małych rozmiarach, moduł roboczy - sensor jako detektor gazu oraz moduł komunikacyjny.

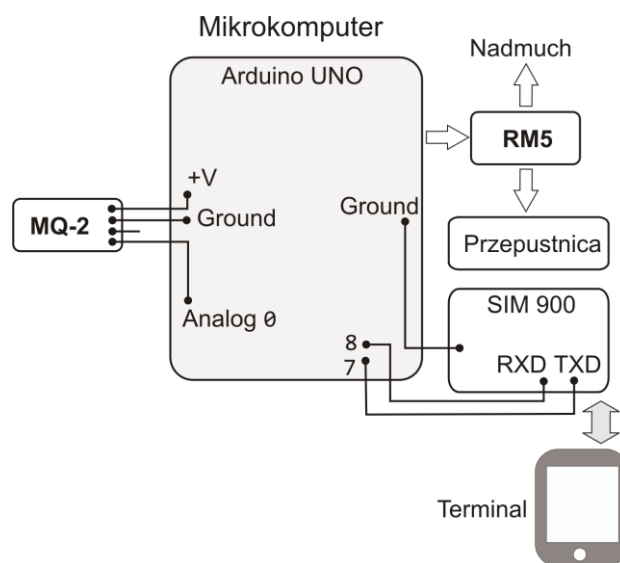
Dysponując mikrokomputerem Arduino wraz z modulem komunikacyjnym SIM oraz wykorzystując infrastrukturę naziemnej sieci komórkowej możemy zaprojektować system kontroli pojazdów z napędzanych gazem LPG lub LNG. Przy obecnie dobrze rozwiniętej architekturze telefonii komórkowej jesteśmy w stanie kontrolować pojazdy z instalacją LPG z dowolnego miejsca. Ulatniający się gaz będzie sygnalizowany nie tylko kierowcy pojazdu ale również dyspozytorom tych pojazdów czy służbom technicznym.

1. SYSTEM DETEKCJI I KONTROLI ULATNIAJĄCEGO SIĘ GAZU

System detekcji, kontroli i monitoringu ulatniającego się gazu składa się z jednostki centralnej, modułów roboczych oraz terminala abonenckiego. Jednostkę centralną stanowi jednopłytkowy mikrokomputer Arduino, sprawuje on kontrolę i nadzór nad modułami roboczymi, steruje nimi, gromadzi i analizuje dane oraz komunikuje się z użytkownikiem. Komunikacja z użytkownikiem odbywa się za pomocą modułu SIM 900 na łączach telefonii komórkowej (rysunek 1).

Moduły to niezależne urządzenia elektroniczne, cechujące się wykonywaniem określonych zadań. Modułami roboczymi są urządzenia elektroniczne pełniące role pomiarowe; pomiar temperatury, wykrywanie gazu w powietrzu i wiele innych. W modelu do detekcji

gazu wykorzystano przetwornik MQ-2. Modułami roboczymi są również interfejsy komunikacji bezprzewodowej oraz urządzenia wykonawcze (przełącznik, RM-5), które załączają inne urządzenia przeciwdziałające skutkom, nawiew powietrza, odcięcie dopływu paliwa itp.

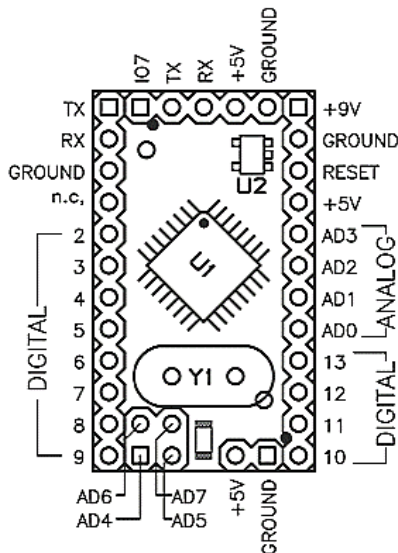


Rys. 1. Elementy systemu detekcji, kontroli i monitoringu ulatniającego się gazu

1.1. Arduino, jednostka centralna systemu

Centralnym podzespołem systemu jest mikrokomputer Arduino UNO (rysunek 2).

Układy oparte na Arduino działają samodzielnie, jako niezależne urządzenia lub też we współpracy z komputerami stacjonarnymi lub przenośnymi. Arduino to 8 bitowy mikrokontroler AVR firmy Atmel. Charakteryzujący się pamięcią rzędu od 4 do 256 kB oraz rozszerzonym zbiorem instrukcji. Większość układów posiada 16 MHz rezonator kwarcowy. Spotykane są również płytki o wyższym lub nieco niższym taktowaniu zegara. W urządzeniach stosuje się trzy rodzaje pamięci: Flash, EEPROM i SRAM.



Rys.2. Piny Arduino UNO

Arduino wyposażone są w złącza analogowe (A0 - A3) i cyfrowe, stanowiące wejścia/wyjścia, umożliwiające podłączenie modułów roboczych stanowiących urządzenia peryferyjne. Piny analogowe współpracują z urządzeniami generującymi napięciowe sygnały analogowe, temperatura, stężenie gazu w powietrzu, itp. Sygnały odbierane na złączach analogowych są rozpoznawalne w zakresie od 0 do 5V i odczytywane w formie dziesiętnej od 0 do 1023 (1024 poziomy kwantyzacji). Wejścia cyfrowe określają zdarzenie, np. wciśnięcie, wyciśnięcie przycisku. Identyfikacja polega na odczytaniu stanu pinu, wykorzystanego do podłączenia układu peryferyjnego. Stan może przyjąć dwie wartości: HIGH – wysoki, LOW – niski. Ponadto możemy odczytać informację o długości trwania danego stanu. Dzięki takiemu zestawowi danych, możemy podejmować określone akcje wewnątrz programu, które będą skorelowane z czynnikami zewnętrznymi. Piny analogowe możemy przekształcić na cyfrowe, technika przekształcania jest prosta.

Wyjścia cyfrowe podobnie jak i wejścia cyfrowe przyjmują stan wysoki oraz niski, jednakże ich wykorzystanie może być bardziej złożone. Za pomocą pinów wyjść cyfrowych jesteśmy w stanie przesyłać wiele informacji w transmisji szeregowej, dzięki temu możliwe jest sterowanie kilkoma urządzeniami za pomocą wielu poleceń pochodzących z tego samego wyjścia (pinu).

Arduino zasilane jest napięciem od 5V DC do 12V DC, z portu USB, a także z dodatkowego gniazda 3,5mm. Port szeregowy USB odpowiedzialny jest także za komunikację Arduino z komputerem. Umożliwia nie tylko odczyt informacji pochodzących z analizy sygnałów wejścia, ale także wczytywanie programu do pamięci mikrokontrolera.

Arduino jest programowalne w środowisku Wiring. IDE (*ang. Integrated Development Environment*), działa jako symulator terminala szeregowego. Środowisko jest bezpłatne i umożliwia łatwe programowanie mikrokontrolera. IDE jest dostępne pod systemy operacyjne Windows, Linux, Mac OS X.

Po zainstalowaniu IDE i uruchomieniu na ekranie komputera (port COM1) wyświetlane jest okno z zakładkami jako menu narzędziowe.

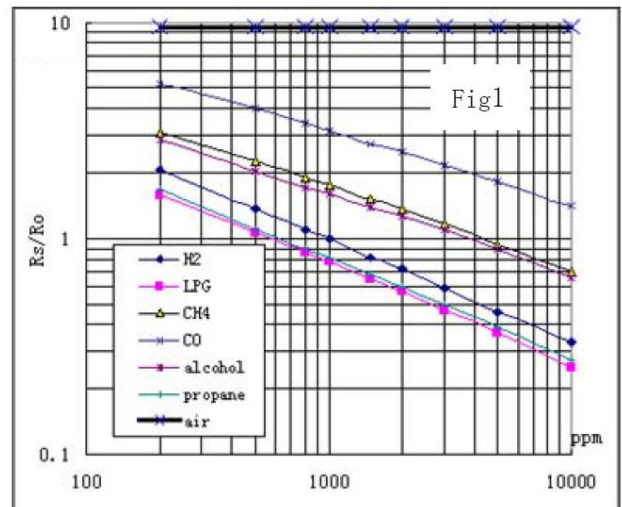
Mikrokontroler jest wstępnie zaprogramowany programem rozruchowym (*ang. bootloader*), i dlatego nie jest wymagane posiadanie zewnętrznego programatora.

Produkty Arduino występują jako gotowe zmontowane układy lub też można zmontować je samodzielnie na podstawie ogólnie dostępnych projektów oraz części i podzespołów elektronicznych.

Środowisko IDE dla Arduino posiada kilkanaście wbudowanych bibliotek. Można również implementować biblioteki obsługujące moduły robocze. W projekcie wykorzystano bibliotekę zewnętrzną SoftwareSerial.h, obsługującą moduł GSM SIM900.

1.2. Moduł detekcji gazu

W systemie detekcji i monitoringu wykorzystano sensor MQ-2, jest to przetwornik półprzewodnikowy, w którym na skutek oddziaływania gazu zmienia się powierzchniowa rezystancja półprzewodnika. Zmiana rezystancji półprzewodnika jest wynikiem reakcji powierzchniowej -dysocjacji jonowej w wyniku, której następuje uwolnienie elektronów, a to prowadzi do zmiany rezystancji roboczej sensora (rysunek 3). Przy braku gazu w otoczeniu sensora stosunek rezystancji R_s/R_0 jest stały i wynosi 9,8 (rysunek 3). Oś odciętych zawiera stężenie gazu w ppm, czyli liczby cząsteczek na milion.



Rys. 3. Charakterystyka zmiany oporności względem stężenia gazu dla sensora MQ-2,

źródło:

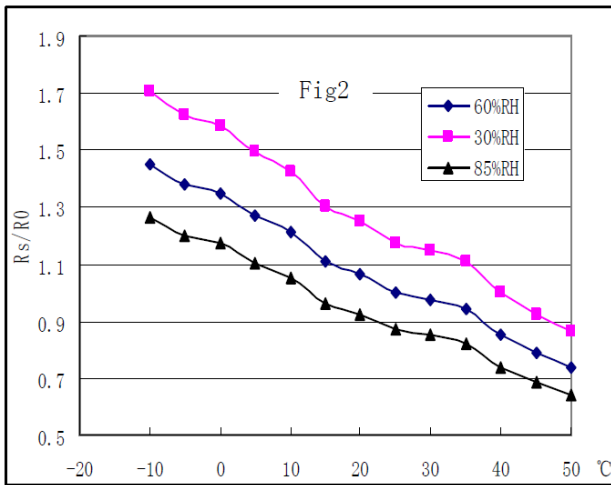
https://www.pololu.com/file/download/MQ2.pdf?file_id=0J309

R_s zmienia się w granicach od 2 k Ω do 20k Ω w zależności od stężenia gazu, natomiast R_0 to wykalibrowana wartość rezystancji przy braku gazu lub jego obecności o bardzo małym natężeniu. Z wykresu widzimy, że stosunek R_s/R_0 maleje przy wzroście stężenia gazu. Duży wpływ na poprawność odczytów pochodzących z sensora, a także działania systemu ma temperatura i wilgotność otoczenia. Przy zmianie temperatury i wilgotności zmienia się również stosunek rezystancji R_s/R_0 (rysunek 4). Okazuje się, że dla mniejszej wilgotności odczyty pochodzące z sensora są niższe niż dla wilgotności większych.

Kalibracja sensora MQ-2

Czujnik MQ-2 jest układem elektronicznym, na wyjściu którego pojawia się odpowiedni poziom napięcia (sygnału), w zależności od stężenia gazu w jego otoczeniu. Stężenie gazu w powietrzu wyrażane jest w jednostkach ppm (liczba cząstek na milion). Jak pokazano na rysunku 3 w czujniku tym rezystancja robocza ulega zmianie w zależności od ppm (100 do 10000 ppm), generując różne poziomy napięcia na elektrodzie wyjściowej. Gdybyśmy czujnik zanurzyli całkowicie w gazie, wówczas stężenie gazu przekracza zakres działania czujnika i czujnik generował by bliżej nie określony sygnał, natomiast małe stężenie, poniżej 100ppm nie powoduje żadnych zmian w układzie. Jak wiadomo sygnał na wyjściu zależy również od warunków atmosferycznych (temperatury i wilgotności), dlatego też sensor MQ-2 (czujnik) należy wyskalować do konkret-

nego projektu, czyli do warunków panujących w miejscu jego montażu. W projekcie MQ-2 skalowano na bazie charakterystyki rezystancyjnej (rysunek 4) przyjmując, że będzie on zainstalowany w otoczeniu z dodatnimi temperaturami w granicach 20 stopni Celsjusa. Stanowi to niewątpliwie problem i zakłopotanie w wyszukaniu takiego miejsca w pojeździe z instalacją



Rys. 4. Wpływ temperatury i wilgotności otoczenia na wartość poziomów sygnału pochodzących z sensora MQ-2, źródło: https://www.pololu.com/file/download/MQ2.pdf?file_id=0J309

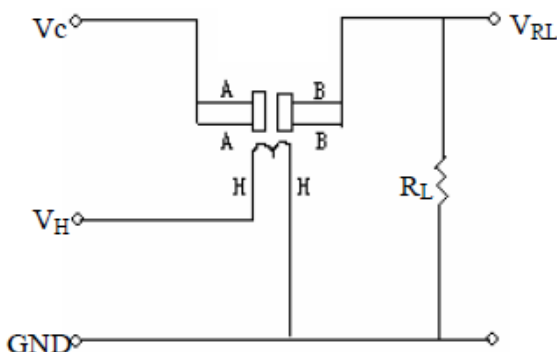
Wyznaczenie optymalnej wartości ppm gazu w ośrodku do wystartowania systemu nie było takie oczywiste. Najpierw przy braku gazu przechwycono sygnał i wyznaczono R_0 , bazując na charakterystyce $R_s/R_0 = f(\text{ppm})$:

$R_0 = R_s/9,8$,
gdzie R_s wynosi;

$$\frac{R_L}{R_L - R_s} = \frac{U_{RL}}{V_v},$$

$$R_s = \frac{V_c}{V_{RL} - 1} \cdot R_L \quad (1)$$

V_c - napięcie zasilające, nie większe niż 24V,
 V_{RL} - napięcie robocze, wyjściowe, zależne od stanu środowiska,
 R_L - rezystor obciążeniowy.



Rys. 5. Układ elektroniczny sensora MQ-2, źródło: https://www.pololu.com/file/download/MQ2.pdf?file_id=0J309

Należy zaznaczyć, że charakterystyka zmian oporności względem stężenia gazu jest w skali logarytmicznej, a to wymusza wyznaczenia nachylenia krzywej:

$$a = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

Weźmy dwa punkty krańcowe:
 $x_2 = 10000$, $x_1 = 200$, $y_2 = 1,6$, $y_1 = 0,26$, wówczas;
 $x_2 = \log 10000 = 4$, $x_1 = \log 200 = 2,3$, $y_1 = \log 1,6 = 0,2$, $y_2 = \log 0,26 = -0,58$,
to;
 $a = -0,46$

Znając nachylenie oraz wybrane współrzędne bazowe x_1 , y_1 na prostej możemy dla dowolnej wartości rzędnej y wyznaczyć wartość odciętej x , czyli odwzorowania rezystancji R_s na stężenie gazu w ppm:

$$y - y_1 = a(x - x_1),$$

$$x = \frac{y - y_1}{a} + x_1.$$

Jak już wspomniano osie charakterystyki sensora przeskalowano na miarę logarytmiczną (decybelową), w związku z tym:

$$\log x = \frac{\log(y - y_1)}{a} + x_1,$$

$$x = 10^{\frac{\log(y - y_1)}{a} + x_1}, \quad (3)$$

x - stężenie gazu w ppm,
 $y = R_s/R_0$.

Załóżmy teraz, że skanowany gaz daje nam stosunek rezystancji R_s/R_0 na poziomie 0,85, korzystając z zależności (3) oraz wybranych punktów charakterystycznych: $x_1 = 2,3$ (200 ppm), $y_1 = 0,2$ (1,6) i nachylenie $m = -0,46$ otrzymamy;

$$x = 10^{\frac{\log(0,85 - 0,2)}{-0,46} + 2,3} = 10^{0,4 + 2,3}$$

$x = 500$ ppm.

1.3. Moduł SIM900

W systemie wykorzystano Moduł SIM90, jest on wykorzystany do wysyłania komunikatów ostrzegawczych o ulatniającym się gazie poprzez łącza telefonii komórkowej w standardzie 3G. Moduł ten współpracuje z Arduino poprzez własną szynę TX RX, którą podpinamy do dowolnych pinów cyfrowych mikrokomputera. Do komunikacji modułu z jednostką centralną wymagana jest gotowa biblioteka SoftwareSerial.h, którą należy pozyskać z sieci od dystrybutora lub producenta modułu. Biblioteka ta jest spakowana w formacie zip i instaluje się ją poprzez zakładkę Szkielet -> Dołącz bibliotekę .zip.

Moduł komunikuje się w oparciu o komendy AT (standard telefonii komórkowej GSM). Każda komenda rozpoczyna się prefiksem AT, następnie znak + a po nim komenda w postaci CXXX. W projekcie wykorzystano tylko dwie komendy:

AT+CMGF = N,

AT+CMGS = adres_przeznaczenia.

Pierwsza komenda umożliwia ustawienie trybu nadawania i odbioru wiadomości SMS. Większość aparatów pracuje w trybie PDU oznaczonym jako 0, pozostałe mają możliwość pracy w trybie TEXT (1). Jeśli w trybie TEXT polecenie wymaga podania numeru abonenta, do którego adresowany jest komunikat.

Linie tekstowe wprowadzane jako treść komunikatów SMS - w trybie tekstowym, kończone są przez znak CTRL+Z. Wprowadzanie każdej komendy można przerwać wysyłając znak CTRL+X.

Komendy AT wysyłane są szeregowo, w związku z tym urządzenia (moduł, jednostka centralna) muszą komunikować się programowalnym portem szeregowym, posiłkując się biblioteką SoftwareSerial lub pochodną.

Prędkość transmisji pomiędzy modulem SIM900 a Arduino wynosi 19 200 bodów - SIM900.begin(19200).

1.4. Programowalna integracja modułów systemu

Współdziałanie modułów i działanie całego systemu przedstawiono w formie programu komputerowego rozbitego w postaci dwóch listingów. Listing 1 zawiera kod programu przetwarzającego sygnał analogowy pochodzący z sensora pomiarowego natężenia gazu i Listing 2, który zawiera kod komunikacji systemu z mobilnym użytkownikiem oraz kod dla komunikatów informacyjnych, przesyłanych użytkownikowi mobilnemu na wybrany numer abonencki.

Listing 1. Program zliczający stężenie gazu w jednostkach ppm

```

1 #define PIN_MQ2      A0
2 #define RL_VALUE    5
3 #define RO_BRAK_LPG  9.83
4 #define GAS_LPG      \\\
5 float LPG_CHAR[3] = {2.3,0.21,-0.47};
6 float Ro = 10;      //Ro w kilo ohms
7 float KaliMQ(int mq_pin)
8 {
9   int i;
10  float value=0;
11  for (i=0;i<50;i++) {
12    value += Kali_r_MQ(analogRead(mq_pin));
13    delay(500);
14  }
15  value = value/50;
16  value = value/RO_BRAK_LPG;
17  return value;
18 }
19
20 float Kali_r_MQ(int raw_adc)
21 {
22  return (((float)RL_VALUE*(1023-raw_adc)/raw_adc));
23 }
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62

```

```

49 Serial.print( "ppm" );
50 delay(200);
51 }
52
53
54
55
56
57
58
59
60
61
62

```

Podajmy analizie powyższy kod z listingu 1. W linii nr 1 zdefiniowano połączenie elektrody sygnałowej MQ-2 z mikrokomputerem. W linii 2 wielkość rezystancji R_L w kilo omach, jest to standardowa rezystancja wbudowana jako dzielnik oporowy (rysunek 5). W kolejnej linii zdefiniowano stosunek dwóch rezystancji (R_s/R_0) w czystym powietrzu, bez oddziaływania gazu. W linii nr 5 parametry do przeliczenia stężenia gazu (wyrażenie 3), bazując na charakterystyce sensora MQ-2. Takie parametry jak: x_1 , y_1 , a , zapisano w tablicy jednowymiarowej jako elementy tej tablicy. Stosunek R_s/R_0 będzie liczony funkcjami i po policzeniu dopiero wraz z elementami tablicy zostaną użyte do wyliczenia stężenia gazu.

W liniach od 7 do 18 zdefiniowana jest funkcja o nazwie **KaliMQ**, która przelicza wartość rezystancji w czystym powietrzu. W pętli pobierane jest 50 próbek. Próbkę pobierane są co pół sekundy. Następnie wyliczona jest średnia z próbek, która dzielona jest przez 9.83.

Kolejną funkcją jest **Kali_r_MQ**, czujnik i rezystor obciążeniowy tworzą dzielnik napięcia. Funkcja wylicza rezystancję czujnika, bazując na napięciu na rezystorze obciążenia (pinie) oraz na jego rezystancji. Ta rezystancja zmienia się na skutek oddziaływania gazu. Argument **raw_adc** stanowi napięcie na wejściu mikrokomputera, produktem wyjściowym jest wyliczona rezystancja czujnika.

W liniach od 24 do 34 zdefiniowana jest funkcja o nazwie **RS_MQ**. Funkcja ta wykorzystuje wcześniejszą funkcję **Kali_r_MQ**, pobiera kilka próbek rezystancji R_s a następnie ją uśrednia.

Funkcja **Gaz_sort** ma znaczenie i sens jeśli system jest zdolny do wykrywania innych gazów i zanieczyszczeń niż gaz LPG, wówczas należy wzbogacić instrukcję warunkowego wyboru „if”.

Ostatnią funkcją w listingu jest **Gaz_krzywa**, przelicza R_s/R_0 na wartość ppm bazując na wyrażeniu (3). Argument **rs_ro_ratio** zawiera R_s/R_0 , natomiast **p_krzywa**, punkty krzywej zapisane w tablicy LPG_CHAR.

Kolejny listing zawiera kod programu realizujący wywołanie wybranego abonenta telefonii komórkowej i przekazanie mu komunikatu ostrzegawczego oraz informacji o stężeniu ulatniającego się gazu.

Listing 2. Program wysyłający komunikat na bazie danych pomiarowych pochodzących z sensora MQ-2

```

1 #include <SoftwareSerial.h>
2 SoftwareSerial SIM900(7, 8);
3 #define PIN_MQ2 A0;
4 void setup()
5 {
6   SIM900.begin(19200);
7   Serial.begin(9600);
8   SIM900power();
9   delay(20000);
10 }

```

```

11 void SIM900power()
12 {
13   digitalWrite(9, HIGH);
14   delay(1000);
15   digitalWrite(9, LOW);
16   delay(5000);
17 }
18 void sendSMS()
19 {
20   SIM900.print("AT+CMGF=1\r");
21   delay(100);
22   SIM900.println("AT + CMGS = \"+48600663426\"");
23   delay(100);
24   SIM900.println("Gaz sie ulatnia");
25   delay(100);
26   SIM900.println((char)26);
27   delay(100);
28   SIM900.println();
29   delay(600);
30   SIM900power();
31 }
32 void loop()
33 {
34   Serial.println(PIN_MQ2);
35   //Wypisanie wartosci odczytu czujnika
36   if(analogRead(PIN_MQ2)>500){
37     // Warunek wysłania wiadomości
38     Serial.println("Gaz sie ulatnia");
39     sendSMS();
40     // Wywołanie funkcji wysłania
41   }
42 }

```

W linii 9 następuje oczekiwanie na zarejestrowanie karty SIM w sieci GSM, czyli powiązanie karty z siecią. Funkcja **SIM900power()** - linia 11 obsługuje przycisk power na module GSM Shield. W linii 20 ustawiono tryb pracy a w 22 połączenie z konkretnym abonentem pod wskazany numer. W linii 24 zamieszczono treść wiadomości SMS do wysłania. Zakończenie komend AT zrealizowano znakiem „Z” tablicy ASCII (char)26. Na wysłanie wiadomości SMS przeznaczono 10 minut (linia 29). Pobrano również wartość natężenia gazu, na pinie A0 i porównano z ustalonym progiem (linia 35). Na koniec w linii 37 wywołano funkcję transmisyjną **sendSMS()** realizującą transmisję danych.

PODSUMOWANIE

Projektowanie systemów na mikroprocesory czy mikrokomputery nie jest skomplikowane. Problem sprowadza się do dobrania odpowiednich modułów roboczych i odpowiedniego zaprojektowania mikrokomputera pod sprawowane funkcje. Przedstawione rozwiązanie monitoringu pokazuje nieograniczony i bogaty potencjał innowacyjności w konstruowaniu systemów i układów do automatycznego sterowania i zarządzania procesami. Podstawowymi elementami systemu są moduły robocze, których zadaniem jest wykonywanie zaprogramowanych czynności i przesłanie danych do jednostki

centralnej i użytkownika końcowego oraz mikrokomputer stanowiący jednostkę centralną. Kontrolę nad pracą modułów roboczych sprawuje moduł centralny..

Niniejsza publikacja może posłużyć studentom oraz inżynierom w projektowaniu automatyki przemysłowej, również w motoryzacji, w robotyce, wszelkich innych rozwiązaniach związanych z cyfrowym przetwarzaniem sygnałów oraz z automatycznym zdalnym sterowaniem procesami.

BIBLIOGRAFIA

1. Wszelak S., *Inteligentne zarządzanie obiektami z wykorzystaniem mikrokontrolerów SBM i SBC (Arduino i Raspberry Pi)*. Logistyka 2014, nr 6.
2. Wszelak S., *Bezprzewodowe usługi sieciowe z wykorzystaniem mikrokomputera Raspberry Pi*. Techniki Transportu Szynowego 2015, nr 12.
3. Wszelak S., *System zarządzania i monitorowania mikroklimatem w naczepach i kontenerach transportowych*, Technika Transportu szynowego 2016, nr 12.
4. <http://layer.uci.agh.edu.pl/pl/dydaktyka/lab-sens/cw4.html> [maj 2017] .
5. <http://systemy.instalacyjne.pl/?p=3682> [lipiec 2017].
6. <http://gazopomiar.pl/download/DOK-348%20kat.pdf> [lipiec 2017].
7. <http://exgaz.pl/charakterystyka-czujnikow/> [lipiec 2017].

Abstract

The paper presents an application for the SIM900 module in SBC microcomputers to monitor vapor leaks in LPG powered vehicles. The paper looks specifically at programming microcomputers to communicate with vapor leaks sensors and with the 3G wireless communication unit. It illustrates numerous applications of programming SBC microcomputers in industrial and automotive automation. More information on the programming environment for microcomputers, including code libraries, involved in program compilation might be found in [1], [2], [3].

Autorzy:

dr inż. **Stanisław Wszelak** – Państwowa Wyższa Szkoła Zawodowa we Włocławku, Instytut Nauk Społecznych i Technicznych; 87-800 Włocławek; ul. 3 Maja 17. Tel. 600663426, e-mail: stanislaw@wszelak.com, www.wszelak.com

JEL: L62 DOI: 10.24136/atest.2018.177

Data zgłoszenia: 2018.05.24 Data akceptacji: 2018.06.15