

Marcin SKÓRA*

PROSTY INTERFEJS DO WIZUALIZACJI DANYCH POMIAROWYCH Z CZUJNIKÓW CYFROWYCH

W referacie przedstawiono możliwość akwizycji i wizualizacji danych pomiarowych, pochodzących z czujników cyfrowych. W roli urządzenia zbierającego dane pomiarowe zastosowano własny moduł mikroprocesorowy, przesyłający bezprzewodowo odczyty czujnika do systemu komputerowego. Jako czujnik wykorzystany został cyfrowy dwuosiowy akcelerometr AIS226DS komunikujący się z otoczeniem poprzez interfejs cyfrowy SPI. Akwizycja i wizualizacja danych została wykonana za pomocą autorskiej aplikacji napisanej w języku C#. W referacie przedstawiono wskazówki przydatne podczas implementacji tego typu programów. Aplikacja została zastosowana do badań akcelerometru MEMS. Na podstawie zaobserwowanych przebiegów powstał algorytm sterowania kierunkiem i prędkością pojazdu elektrycznego (prototypowego wózka inwalidzkiego), w zależności od pochylecia głowy osoby kierującej.

1. WPROWADZENIE

Spośród elektronicznych układów pomiarowych coraz częściej wybierane są czujniki cyfrowe, zwracające do urządzenia nadrzędnego wynik pomiaru za pomocą różnego typu interfejsów cyfrowych. Daje to możliwość umieszczenia wewnątrz struktury czujnika dodatkowych bloków, np. kompensacji temperaturowej wielkości mierzonych czy linearyzacji charakterystyki. Z drugiej jednak strony brak jest bezpośredniego śledzenia wartości mierzonych, np. za pomocą oscyloskopu lub po prostu woltomierza. W związku z tym powstaje potrzeba konstrukcji układu do komunikacji z czujnikiem za pomocą używanego przez niego interfejsu cyfrowego. Następnie należy wybrać sposób wykorzystania otrzymanego pomiaru, np. czy ma być dostępny jako napięcie analogowe pokazywane na wyświetlaczu (oscyloskopie), czy też przesyłany do komputera.



Rys. 1. Schemat przepływu informacji w prezentowanym układzie pomiarowo-wizualizacyjnym

* Politechnika Wroclawska.

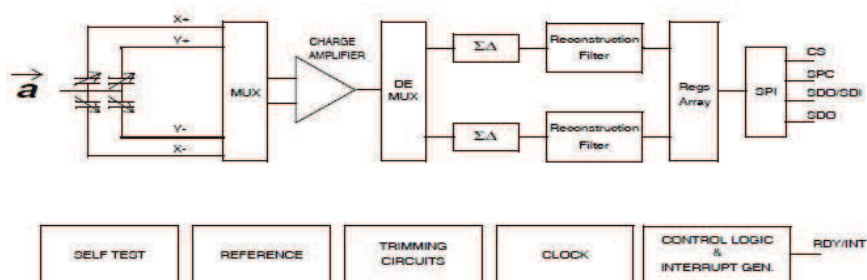
W referacie została wybrana druga możliwość. Na rys. 1 został przedstawiony schemat przepływu informacji w prostym systemie pomiarowo-wizualizacyjnym. Spotykane są rozwiązania tego typu, podobny pomysł można znaleźć np. w [1].

2. MODUŁ CZUJNIKA Z INTERFEJSEM CYFROWYM

2.1. Czujnik z interfejsem cyfrowym

Jako czujnik z interfejsem cyfrowym wybrano układ AIS226DS firmy STMicroelectronics. Jest to dwuosiowy akcelerometr, o zakresie pomiarowym wybieralnym $\pm 2g$ lub $\pm 6g$. Przeznaczony jest na szeroko pojęty rynek samochodowy, a w szczególności do zabezpieczeń antykradzieżowych, pomiarów przesunięcia, monitorowania wibracji i obsługi funkcji aktywowanych ruchem [2]. Układ zamknięty jest w łatwej do montażu obudowie SO16W.

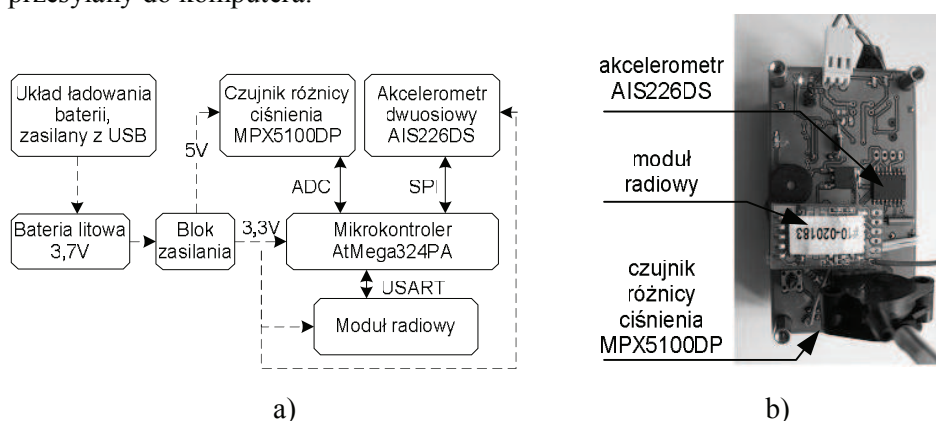
Jeśli chodzi o budowę wewnętrzną czujnika, to powstał on w technologii MEMS. Mierzone przyspieszenie jest pochodną zmian pojemności wewnętrznych kondensatorów z ruchomą okładką (rys. 2). Wynik przechowywany jest z 14-bitową rozdzielczością w rejestrach wewnętrznych.



Rys. 2. Schemat blokowy układu AIS226DS [2]

Czujnik ten umieszczony jest w module pomiarowym (rys. 3), w którym centralną rolę odgrywa mikrokontroler (tu ATmega324PA firmy Atmel). Komunikacja (dostęp do rejestrów akcelerometru) pomiędzy wymienionymi układami zachodzi za pośrednictwem interfejsu SPI (ewentualnie I²C). Zgodnie ze schematem z rys. 1, dane udostępniane są dalszym podmiotom za pośrednictwem łącza szeregowego – modułu USART. Pierwszy moduł radiowy, w roli nadajnika, podłączono do mikrokontrolera, zgodnie z rys. 3. Natomiast drugi, w roli odbiornika, z wyjściem USB, został dołączony do komputera. Komunikują się one między sobą w powszechnie dostępnym paśmie ISM o częstotliwości 868 MHz. Brak przewodów oraz zasilanie bateryjne pozwalają na swobodne manewrowanie modułem z akcelerometrem.

Dodatkowo na płycie drukowanej modułu zbierającego dane znajduje się złącze dla analogowego czujnika różnicy ciśnienia MPX5100DP firmy Freescale Semiconductor. Jego analogowy sygnał wyjściowy jest próbkowany przez wbudowany w układ mikrokontrolera przetwornik analogowy-cyfrowy i również przesyłany do komputera.



Rys. 3. Schemat ideowy (a) oraz wygląd modułu zbierającego dane pomiarowe (b)

2.2. Implementacja interfejsu SPI w mikrokontrolerze

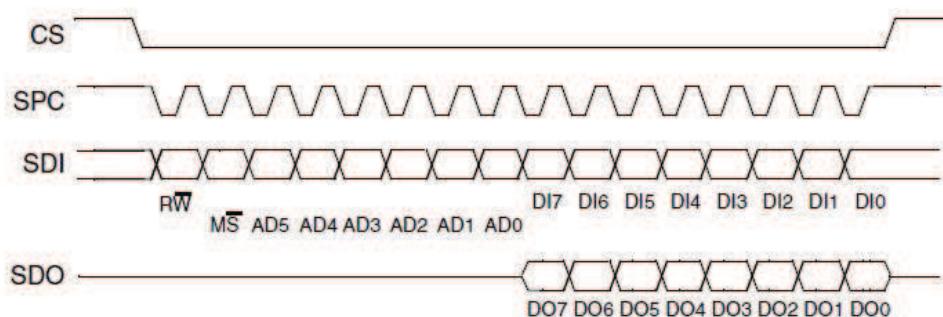
SPI (ang. *Serial Peripheral Interface*) jest szybkim dwukierunkowym interfejsem szeregowym, umożliwiającym połączenie układu nadrzędnego (*master*) z wieloma układami podrzędnymi (*slave*). Do działania wykorzystuje cztery linie: zegarową SCK, wyjście danych z *mastera* do *slave'a* MOSI, wejście danych do *mastera* od *slave'a* MISO oraz linię wyboru *slave'a* \overline{SS} . W systemie może być wiele układów podrzędnych, w danej chwili transmisja danych możliwa jest tylko z jednym z nich, wybranym za pomocą linii \overline{SS} .

Implementacja interfejsu w mikrokontrolerach rodziny AVR ATmega może nastąpić na jeden z kilku sposobów [3]:

- korzystając z wbudowanego sprzętowego modułu SPI – dostępnego w prawie każdym mikrokontrolerze rodziny AVR ATmega,
- korzystając z wbudowanego sprzętowego modułu USI (ang. *Universal Serial Interface*). Jest to moduł elastyczny, dostępny w niektórych mikrokontrolerach (np. ATmega 325V), umożliwiający transmisję w standardach SPI oraz I²C, alternatywnie pracujący jako 4-bitowy licznik.,
- korzystając z modułu USART przystosowanego do pracy w trybie SPI jako *master*. Opcja ta dostępna jest w niektórych układach, w tym ATmega 324PA (rys. 3a). Jej zaletą jest buforowanie transmisji, czego brakuje w zwykłym module SPI.

- w sposób programowy, gdy brak sprzętowego modułu SPI. Rozwiązanie to dodatkowo absorbuje czas mikrokontrolera, potrzebny na wysyłanie i odbiór kolejnych bitów.

Przedstawione wyżej sposoby (oprócz programowej implementacji) stosowano na różnych etapach testowania systemu pomiarowego. Możliwość wyboru pozwala na pewną elastyczność podczas projektowania obwodu drukowanego. Niezależnie jednak, który moduł realizuje transmisję, jej przebieg jest taki sam, dla używanego akcelerometru przedstawiony na rys. 4. Linii CS odpowiada linia \overline{SS} modułu SPI pracującego w trybie *Mode 0*, a jej niski stan logiczny powoduje wybór *slave'a* do transmisji. Sygnał zegarowy SPC pochodzi od układu *mastera*. Liniami SDI są przesyłane informacje od układu nadrzędnego (czy chce on zapisać dane czy odczytać, czy automatycznie zwiększać adres rejestru, adres rejestru oraz właściwe dane, zaczynając od najbardziej znaczącego bitu). Informacja zwrotna zwracana jest poprzez linię SDO.



Rys. 4. Przebieg transmisji z akcelerometrem poprzez interfejs SPI [2]

2.3. Przekazywanie danych

Rola mikrokontrolera w układzie jak na rys. 3a sprowadza się do przekazywania danych od czujników do komputera. Zbiera on wartości odczytane z czujników w jedną spójną ramkę, którą następnie transmituje drogą radiową. W tab. 1 przedstawiono postać wspomnianej ramki. Rozpoczyna się ona dwoma znacznikami startu (liczba 165, dwójkowo 0b10100101), po których następują pola z informacją o ciśnieniu, przyspieszeniu w osi X, w osi Y, bajt statusowy, suma kontrolna i dwa bajty końca ramki (liczba 90, dwójkowo 0b01011010). Ponieważ sygnały na wyjściach z czujników są liczbami dwubajtowymi, przyjęto, że najpierw następuje transmisja bajtu młodszego, później starszego. Suma kontrolna to suma pól danych (od nr 3 do nr 9 w tab. 1), dzielona modulo 256 – jest to intuicyjny i skuteczny sposób na wykluczenie większości błędnych ramek z interpretacji.

Tabela 1. Postać ramki danych przesyłanych drogą radiową

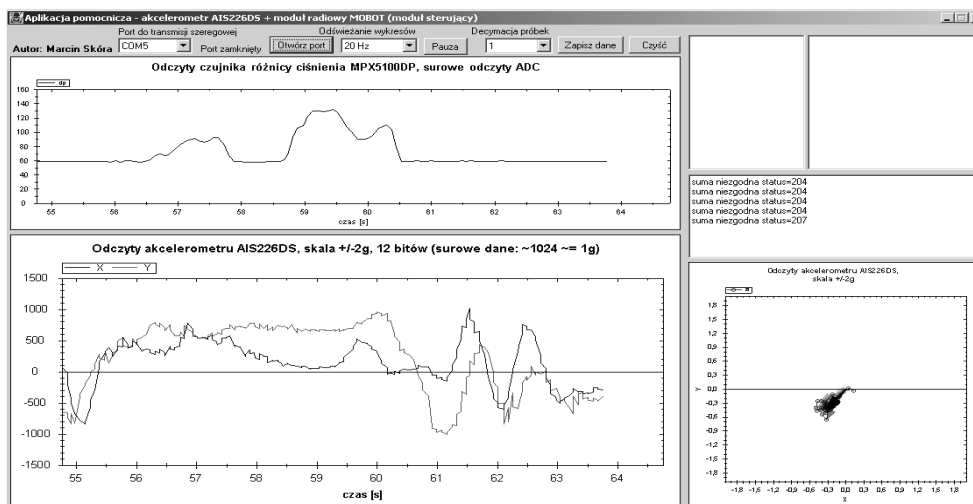
opis	START	START	CIS L	CIS H	XL	XH	YL	YH	STATUS	SUMA	STOP	STOP
nr	1	2	3	4	5	6	7	8	9	10	11	12

3. APLIKACJA KOMPUTEROWA DO WIZUALIZACJI I AKWIZYCJI DANYCH POMIAROWYCH

3.1. Funkcjonalność programu

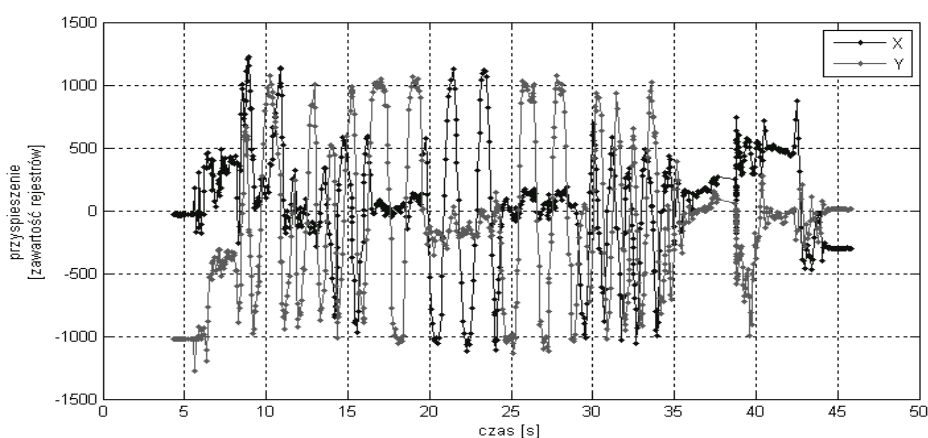
Dane pomiarowe zbierane przez mikrokontroler i przesyłane drogą radiową są przetwarzane przez aplikację użytkownika. Została ona napisana w języku C#. Jest to obiektowy język wysokiego poziomu, który wraz z bibliotekami *.NET Framework* pozwala w prosty sposób tworzyć aplikacje okienkowe, przede wszystkim dla systemu *Microsoft Windows*. W ten sposób czas od powstania pomysłu aplikacji graficznej do jej wykonania zostaje mocno skrócony.

Na rys. 5 przedstawiono wygląd utworzonej aplikacji. Główną część okna zajmują pola wykresów, w tym wypadku zależności przyspieszeń w osiach X oraz Y od czasu, hodograf przyspieszeń (zależność przyspieszenia w osi Y od przyspieszenia w osi X). Dodatkowe okno pozwala na podgląd zależności różnicy ciśnienia, mierzonej przez układ MPX5100DP znajdujący się również w mikroprocesorowym module pomiarowym. W prawym górnym rogu aplikacji z rys. 5 przewidziano miejsce na wyświetlanie ewentualnych komunikatów statusowych lub o błędach, które najczęściej dotyczą różnicy w sumie kontrolnej.



Rys. 5. Wygląd aplikacji do wizualizacji danych pomiarowych

Moduł radiowy podłączony do złącza USB komputera z punktu widzenia programisty udostępnia port szeregowy, którego obsługa jest prosta i szybka. Dlatego też w górnej części interfejsu umieszczono listę dostępnych w systemie portów, by w bezproblemowy sposób nawiązać komunikację. Ponadto górną część aplikacji zajmują kontrolki sterujące wyświetlaniem wykresów (częstotliwością odświeżania, stopniem decymacji próbek) oraz zapisywaniem danych do dalszej obróbki, w formacie CSV oraz jako plik skryptowy programu Matlab, generujący wykresy (rys. 6).



Rys. 6. Przykładowy wykres uzyskany w trakcie działania programu z wszystkich zapisanych danych

3.2. Wskazówki implementacyjne

Aby skrócić czas od pomysłu do realizacji programu, wykorzystano popularną i darmową bibliotekę *ZedGraph* [4]. Służy ona do rysowania wszelkich wykresów statycznych (liniowych, kołowych, histogramów, itp.) i dynamicznych oraz umożliwia także ich bezpośredni zapis do pliku. Zastosowanie sprawdzonego rozwiązania odciąża programistę i pozwala na uniknięcie błędów.

Podczas konstrukcji programu należy rozważyć działania prowadzące do spadku wykorzystania mocy obliczeniowej komputera – poprzez ograniczenie ilości odświeżeń wykresu oraz wykonywanie tego w osobnym wątku, co jest korzystne szczególnie w komputerach z procesorami wielordzeniowymi.

W trakcie działania programu mogą zdarzyć się sytuacje wyjątkowe, np. próba nawiązania połączenia poprzez nieistniejący port szeregowy, odłączenie modułu radiowego podczas pracy programu i szereg innych. Warto zabezpieczyć się na wypadek zaistnienia powyższych sytuacji, by móc przynajmniej zachować dotychczasowe dane przed niespodziewanym zakończeniem programu lub chociaż poznać jego powód. Służy do tego mechanizm wyjątków, tj. bloki `try{} catch{} finally{} umieszczone w odpowiednich miejscach w kodzie.`

4. PODSUMOWANIE

Zaprezentowany prosty interfejs umożliwia podgląd sygnału wyjściowego z czujników cyfrowych w czasie quasi-rzeczywistym (opóźnienie spowodowane jest czasem przesyłania danych, ich obróbki w mikrokontrolerze oraz komputerze oraz częstotliwością odpytywania czujników). Do jego realizacji nie potrzeba skomplikowanego zaplecza laboratoryjnego ani drogiej aparatury, czy komercyjnego oprogramowania. W zupełności wystarczą darmowe programy oraz podstawowe elementy z pracowni elektronicznej.

Do wad przedstawionego systemu zaliczyć trzeba brak uniwersalności – dla każdego dodatkowego czujnika należy zmodyfikować ramkę danych, a w aplikacji komputerowej dodać/zmienić okno wykresu. Ponadto wymagana jest umiejętność pisania programów komputerowych.

Prezentowany w referacie program posłużył autorowi do badań czujników MEMS (na przykładzie akcelerometru z wyjściem cyfrowym). Pozwolił także na powiązanie siły strumienia powietrza wydmuchiwanego z ust z wartościami ciśnienia, mierzonymi czujnikiem różnicowym. Wielkości te posłużyły do opracowania algorytmu sterowania elektrycznego wózka inwalidzkiego, gdzie kierunek jazdy określa pochylenie głowy mierzone akcelerometrem, a pozwolenie na jazdę zachodzi wskutek odpowiedniego dmuchania ustami [5].

LITERATURA

- [1] Örücü S., Tarimer I., Opracowanie kart zbierania danych i oprogramowania do zastosowaniach pomiarowych, *Przegląd Elektrotechniczny*, 85 (2009), nr 10, 166-170.
- [2] Nota katalogowa AIS226DS. Doc ID 15164 Rev. 3, STMicroelectronics.
- [3] Baranowski R., *Mikrokontrolery AVT ATmega w praktyce*, wyd. BTC, Warszawa 2005.
- [4] <http://sourceforge.net/projects/zedgraph>, dostęp: grudzień 2012.
- [5] Skóra M., *Zastosowanie czujników MEMS do sterowania napędu elektrycznego wózka inwalidzkiego*, Praca dyplomowa magisterska, Wrocław 2012.

A SIMPLE INTERFACE FOR DATA VISUALIZATION FROM DIGITAL SENSORS

The paper presents the possibility of acquisition and visualization of the measurement data from digital sensors. The originally microprocessor module was developed, which transmits sensor readings wirelessly to a computer system. The dual axis digital accelerometer AIS226DS was used as a sensor, which communicates with the environment through the SPI interface. Different possibilities of SPI implementation were tested. Data acquisition and visualization were performed using proprietary application written in C#. This paper presents useful guidance in the implementation of such programs. An example application was used to introduce with MEMS sensors. The observed waveforms from the accelerometer allowed to develop the algorithm for control the direction and speed of an electric vehicle (prototype wheelchair), based on the inclination of the head.