

Biblioteka graficzna do sterownika z wyświetlaczem OLED

mgr inż. Marcin Jura
Instytut Techniki Górniczej KOMAG

Streszczenie:

W artykule przedstawiony został proces opracowania biblioteki do sterowników z wyświetlaczem OLED. Przeprowadzono przegląd wybranych bibliotek graficznych. Pozwoliło to na zapoznanie się z ich funkcjonalnością zależną od platformy docelowej i przeznaczenia. Zdobyta wiedza posłużyła do opracowania biblioteki graficznej przystosowanej do wykorzystania w sterownikach. Bibliotekę graficzną przetestowano na Module Inteligentnego Sterownika MIS-1 będącego częścią systemu KOGASTER. Biblioteka zostanie wykorzystana w szczególności w iskrobezpiecznych urządzeniach z wyświetlaczem OLED.

Słowa kluczowe: biblioteka graficzna, wyświetlacz OLED

Keywords: graphics library, OLED display

Abstract:

Process of developing the library for the controller with an OLED display is presented in the paper. Selected graphics libraries are reviewed. Their functionality depending on the target platform and purpose is determined. Acquired knowledge helped to develop graphics library adjusted to work with the controllers. The graphics library is tested on an Intelligent Controller Module MIS-1, a part of KOGASTER system. The library will be implemented in intrinsically safe devices with an OLED display.

1. Wprowadzenie

W ITG KOMAG prowadzone są projekty, w których stosowane są sterowniki iskrobezpieczne własnej konstrukcji. Wykorzystywanie tych sterowników wymaga pisania oprogramowania dedykowanego pod konkretne rozwiązanie. Biblioteka graficzna dedykowana pod wykorzystywany sterownik ułatwi i przyspieszy ten etap pracy.

W zastosowaniach górniczych sterowniki, które są zaprojektowane zgodnie z wytycznymi dyrektywy ATEX oraz normami zharmonizowanymi, muszą spełniać odpowiednie wymagania w zakresie poboru mocy. Większość mocy pobieranej przez sterownik jest wykorzystywana do obsługi wyświetlaczy. Ograniczenie mocy pobieranej przez wyświetlacz znacząco ułatwia projektowanie sterowników iskrobezpiecznych.

Przepisy związane z projektowaniem urządzeń iskrobezpiecznych, poza mocą pobieraną przez wyświetlacz, wpływają również na dostępną moc obliczeniową i dostępną pamięć operacyjną. Ograniczenia w zakresie pamięci operacyjnej wpływają na konieczność minimalizacji objętości oprogramowania. Moduł Inteligentnego Sterownika MIS-1 został wykorzystany jako platforma sprzętowa dla nowej biblioteki graficznej.

Moduł MIS-1 jest częścią rozproszonego systemu [1] KOGASTER. System ten ze względu na jego iskrobezpieczeństwo może być stosowany w systemach sterowania dla maszyn górniczych [12, 13, 17]. Wykorzystywany jest on także przy tworzeniu systemu pozycjonowania kombajnu chodnikowego [9, 10]. Istnieje możliwość jego wykorzystania w systemach z komunikacją bezprzewodową [7, 8], monitoringu i diagnostyki [11] oraz z wykorzystaniem modułów samo zasilających [14, 15, 16].

Obecnie stosowane wyświetlacze w sterownikach są dostarczane przez ich producentów z podstawowym interfejsem. Pozwala to na ustawienie podstawowych parametrów i na przesyłanie ciągu pikseli. Ten niskopoziomowy dostęp do wyświetlaczy stwarza konieczność zaprogramowania obsługi plików graficznych, czcionek i menu, przy każdym nowym zastosowaniu wyświetlacza. Opracowanie odpowiednich bibliotek graficznych pozwala zatem na znaczne przyspieszenie prac związanych z programowaniem obsługi wyświetlacza.

Przeprowadzono przegląd bibliotek graficznych, co pozwoliło to na opracowanie koncepcji obsługi wyświetlacza OLED. Opracowano narzędzie programowe w postaci biblioteki graficznej, stosowanej do obsługi rodziny procesorów PIC32MX5XX/6XX/7XX. Biblioteka napisana w języku C, umożliwi rozbudowaną obsługę wyświetlaczy bazujących na interfejsie SSD1322. Jej funkcjonalność pozwoli na wykonywanie podstawowych i złożonych funkcji graficznych oraz na szybkie tworzenie zaawansowanych interfejsów użytkownika. Zastosowanie opracowanej biblioteki umożliwia łatwiejszy dostęp do funkcjonalności oferowanej przez system KOGASTER [2].

2. Przeznaczenie bibliotek graficznych

Biblioteki graficzne są zbiorami narzędzi, które mają za zadanie ułatwić wyświetlanie grafiki komputerowej. Wiąże się to zazwyczaj z dostarczeniem funkcji podstawowych zadań renderujących. Zadania te polegają na przedstawieniu cyfrowych danych w postaci, która może zostać odczytana przez sterownik wyświetlacza. Renderowanie może być wykonywane na głównym lub na dodatkowym procesorze, dedykowanym do operacji graficznych, który jest wykorzystywany między innymi w komputerach osobistych. Taki zestaw narzędzi pozwala programiście na pominięcie takich zadań, jak: optymalizacja renderowania i części funkcji graficznych, które zawarto w bibliotece.

Biblioteki graficzne, ze względu na ułatwienie pracy nad oprogramowaniem, są wykorzystywane tam, gdzie występuje konieczność wykorzystania wyświetlaczy graficznych. Zdarza się, że wykorzystywane są one również w urządzeniach nie mających bezpośredniego dostępu do wyświetlacza. W takich przypadkach grafika jest zapisywana lub przesyłana do innych urządzeń, które je wyświetlają. Biblioteki graficzne są dostępne na większość dostępnych procesorów i obsługują różne typy wyświetlaczy.

Biblioteki graficzne dostarczają narzędzia na różnym poziomie abstrakcji. Przykładowo, jedna biblioteka może udostępniać funkcje umożliwiające rysowanie prymitywów graficznych, takich jak np. prostokąt, a inna przyciski. W związku z tym wiele bibliotek korzysta z innych, które ułatwiają opracowanie kolejnej. Ma to miejsce najczęściej w urządzeniach o dużej mocy obliczeniowej. Przykładem takich bibliotek bazowych są: OpenGL, DirectX, które bezpośrednio komunikują się z procesorem graficznym. Ich funkcjonalność bazuje głównie na umożliwieniu komunikacji między głównym procesorem (CPU) a procesorem graficznym (GPU).

3. Koncepcja biblioteki graficznej

Opracowanie koncepcji nowej biblioteki graficznej wymagało określenia docelowej platformy sprzętowej oraz przeznaczenia biblioteki. Określenie platformy pozwoliło na przyjęcie dostępnej mocy obliczeniowej, ilości pamięci danych i programu. Natomiast

określenie przeznaczenia biblioteki pozwoli zdefiniować funkcjonalności, które powinny w niej się znaleźć.

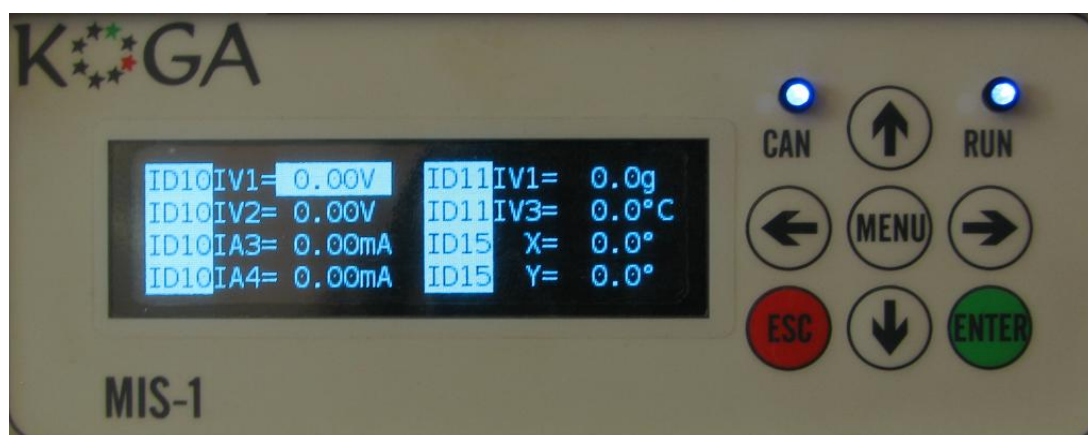
Jako platformę dla biblioteki graficznej przyjęto Moduł Inteligentnego Sterownika MIS-1 systemu KOGASTER (rys. 1) [3], rozwijanego w ITG KOMAG od kilku lat. System jest częścią rozproszonego układu sterowania wykorzystującym magistralę CAN i protokół CANopen.



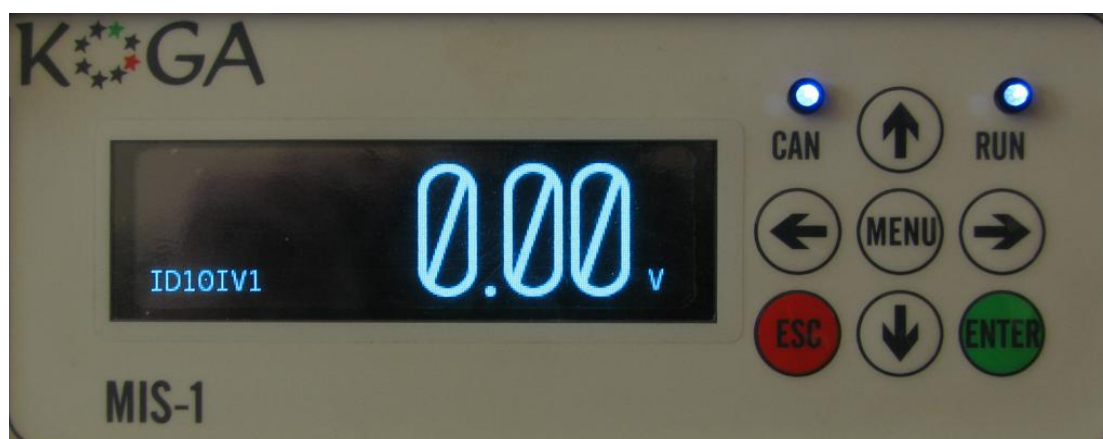
Rys. 1. Moduł Inteligentnego Sterownika MIS-1 [4]

Moduł MIS-1 (rys. 1) wykorzystuje mikrokontroler PIC32MX795F512L z 512 KiB pamięci programu oraz 128 KiB pamięci RAM. W module MIS-1 stosowany jest wyświetlacz wykorzystujący kontroler SSD1322, który pozwala na komunikację poprzez magistralę I2C lub SPI.

W celu wyznaczenia najistotniejszych funkcjonalności biblioteki graficznej, utworzono koncepcję interfejsu graficznego (GUI) użytkownika w odniesieniu do MIS-1. Koncepcję oparto o sterownik MIS-1 zastosowany w kolejce szynowej. Główny ekran (rys. 2) zaprojektowano tak, by wyświetlane były wszystkie zmienne dostarczane przez moduły podłączone do MIS-1. Pozwoliło to na stały podgląd wszystkich istotnych parametrów pracy monitorowanego obiektu. Zaprojektowano również ekran (rys. 3), który umożliwił wyświetlanie pojedynczej zmiennej dużą czcionką wypełniającą cały obszar wyświetlacza. Opracowano strukturę menu (rys. 4), która pozwoliła na obsługę dodatkowych funkcjonalności.



Rys. 2. Ekran z wyświetlanymi zmiennymi [6]



Rys. 3. Ekran z wyświetlaną pojedynczą wartością [6]



Rys. 4. Fragment menu umożliwiający regulację podświetlenia [6]

4. Wybór narzędzi programistycznych

Środowisko programistyczne MPLAB X IDE przeznaczone do pisania oprogramowania i programowania typoszeregu procesorów PIC pozwala na wykorzystanie języka C i C++. Język C++ bazujący na C jest językiem opartym na paradygmatach programowania proceduralnego i strukturalnego. C++ jest rozszerzone dodatkowo o paradygmat programowania obiektowego i uogólnionego, co umożliwia tworzenie fragmentów kodu, którego wielokrotne użycie jest ułatwione. Ułatwia on pracę programistów i pozwala na tworzenie bardziej przejrzystego kodu. Język C jest bardzo popularny przy zastosowaniu w mikroprocesorach z małą ilością pamięci i niską częstotliwością taktowania, co wynika z mniejszego stopnia jego skomplikowania. Język C++ pozwala na wykorzystanie bibliotek napisanych w języku C lub zastosowania fragmentów kodu w nim napisanych.

Ze względu na mniejsze zapotrzebowanie zasobów procesora i możliwości wykorzystania biblioteki w oprogramowaniu napisanym w C lub C++ przy tworzeniu biblioteki zdecydowano się skorzystać z języka C. Wybór tego języka da większą elastyczność zastosowania dla użytkowników biblioteki. Zdecydowano, że jedną z podstawowych funkcjonalności biblioteki będzie wyświetlanie tekstu. Dodatkowo przewidziano obsługę plików graficznych.

5. Oprogramowanie

Przy opracowywaniu biblioteki powstało oprogramowanie pozwalające na przetestowanie i demonstrację biblioteki, oraz skrypty mające na celu konwersję plików do formatów wspieranych przez powstałą bibliotekę graficzną.

Funkcje udostępnione przez bibliotekę dla użytkownika pozwalają na:

- wyświetlanie pojedynczego piksela w jednym z szesnastu odcieni szarości,
- odwracanie odcieni szarości na wyznaczonym obszarze, w kształcie prostokąta,
- wyświetlanie wybranej ikony z bitmapy, w wyznaczonym miejscu,
- wyświetlanie pojedynczego znaku z wybranej czcionki i w wybranym miejscu,
- wyświetlanie ciągu znaków z wybranej czcionki i w wybranym miejscu,
- wyświetlanie liczby zmiennoprzecinkowej za pomocą czcionki o wysokości ekranu i w wybranym miejscu.

Udostępniono także strukturę, która umożliwia łatwe i intuicyjne tworzenie menu o strukturze drzewa. Wykorzystanie tej struktury pozwala na stworzenie menu z zagnieżdżeniami.

Ze względu na wymiary wyświetlacza dostępnego w MIS-1 zdecydowano się na użycie dwóch rozmiarów czcionek z pełnym alfabetem. Wybrano wysokości czcionki 10 i 16 pikseli, które pozwalają na zapisanie na ekranie odpowiednio 6 i 4 linii tekstu. Ze względu na ograniczoną ilość dostępnej pamięci zdecydowano się użyć znaków ASCII, poszerzonych o polskie znaki i znak stopnia. Zakres znaków ustalono na potrzeby wcześniej stworzonej koncepcji interfejsu graficznego. Dodatkowo zdecydowano się dodać czcionkę zawierającą tylko liczby i separator dziesiętny, której wysokość pokrywałaby się z wysokością ekranu. Zaplanowano również użycie ikon, w celu komunikacji potencjalnych zagrożeń i oznaczenia wybranych funkcjonalności interfejsu graficznego.

Opracowano trzy czcionki, zgodne z przyjętymi założeniami. Pierwotnie do konwersji czcionki wykorzystano aplikację webową, która pozwalała na przekształcenie bitmap na tablice bajtowe. Okazało się jednak, że niemożliwe jest przechowanie wszystkich czcionek na mikrokontrolerze. W celu zmniejszenia ilości przechowywanych danych opracowano skrypt, który umożliwił zapis dwóch pikseli na jednym bajcie. Dostępny wyświetlacz ma możliwość wyświetlania w 16 odcieniach szarości. Binarnie można taki odcień zapisać w 4 bitach, co w konsekwencji zmniejszyło rozmiar zapisanych danych o połowę.

W celu wspierania wszystkich dostępnych znaków w zaimplementowanych czcionkach zdecydowano się skorzystać z kodowania UTF-8, w celu zapisu ciągów znaków. W skrypcie umieszczono funkcję pozwalającą na zapis ciągów znaków zakodowanych w UTF-8, do postaci tablicy bajtowej. Tak zakodowana tablica pozwoli na wykorzystanie ciągów bez konieczności zmian kodowania w zintegrowanym środowisku programistycznym. Umożliwi także uniknąć ograniczeń wprowadzonych przez środowisko, które nie pozwala na zastosowanie różnego kodowania znaków w jednym projekcie.

Przygotowano zestaw ikon (rys. 5), który umożliwia prezentację różnych stanów awaryjnych. Możliwe jest złożenie całej ikony z kilku fragmentów. Oznacza to, że przy rysowaniu grafiki nie jest ona nadpisywana na istniejącą, a jest dokonywana operacja sumy binarnej. Pozwala to na nakładanie na siebie kilku ikon. Ikony złożone z kilku grafik przedstawiono w drugim wierszu na rysunku 6.

Pliki graficzne zapisane w mikrokontrolerze, nawet po kompresji, zajmowały znaczącą część dostępnej pamięci. Utworzono zatem bitmapę (rys. 7), która zawiera wszystkie kształty potrzebne do stworzenia wymaganych znaków. Funkcja wykorzystująca tę grafikę została napisana tak, by możliwe było tworzenie znaków przy pomocy nakładania na siebie fragmentów bitmapy i funkcji do rysowania prostokątów. Złożone w ten sposób znaki przedstawiono na rysunku 6.



Rys. 5. Grafika z ikonami [6]



Rys. 6. Ikony przedstawiane na wyświetlaczu [6]



Rys. 7. Grafika umożliwiająca dynamiczne tworzenie liczb o wysokości całego ekranu [6]

Na podstawie opracowanej koncepcji interfejsu graficznego oprogramowanie MIS-1 rozbudowano o graficzny interfejs użytkownika. Do obsługi interfejsu graficznego wykorzystano przyciski znajdujące się po prawej stronie wyświetlacza. Wybór pojedynczej wartości do wyświetlenia (rys. 3) odbywa się poprzez wykorzystanie przycisków ze strzałkami. Aktualnie zaznaczona wartość wskazywana jest przez inwersję koloru. Przejście do ekranu na rysunku 3 odbywa się przez klawisz „ENTER”, a powrót przez klawisz „ESC”.

W celu przejścia do menu wykorzystywany jest klawisz „MENU”. Menu wyświetlane jest w postaci listy, która przedstawia tę gałąź struktury menu, w której aktualnie znajduje się użytkownik. Do przemieszczania się po menu wykorzystywane są przyciski ze strzałkami w górę, w dół, „ENTER” i „ESC”. Pierwsze dwa pozwalają na przesuwanie zaznaczenia odpowiednio w górę i w dół. Przycisk „ENTER” wykorzystywany jest do wchodzenia w głąb struktury menu lub zaznaczenia docelowej opcji. Przycisk „ESC” natomiast wykorzystywany jest do przejścia do gałęzi nadrzędnej.

6. Podsumowanie

Przeprowadzony przegląd istniejących bibliotek graficznych pozwolił na zapoznanie się z ich funkcjonalnością. Biblioteki przeanalizowano pod kątem ich docelowej platformy i przeznaczenia. Na tej podstawie opracowano koncepcję biblioteki graficznej.

Jako urządzenie spełniające założenia koncepcji przyjęto Moduł Inteligentnego Sterownika MIS-1, będącego częścią systemu KOGASTER [4]. Na podstawie możliwości jego wykorzystania zaprojektowano działanie przykładowego interfejsu graficznego użytkownika w innowacyjnym systemie mechatronicznym [5]. Wizja tego interfejsu pozwoliła na określenie najbardziej potrzebnych funkcjonalności zawartych w bibliotece graficznej.

Powstała biblioteka, w której zawarto:

- podstawowe funkcje graficzne (np. rysowanie prymitywów, inwersja obrazu),
- obsługa plików graficznych,
- obsługa czcionek,
- tworzenie menu o strukturze drzewa.

W trakcie tworzenia biblioteki zaistniała potrzeba przetwarzania danych w celu zmniejszenia zapotrzebowania na pamięć i zapewnienia zgodności z kodowaniem. Zmniejszenie wykorzystywanej pamięci zostało uzyskane dzięki skryptowi, który pozwolił na zmniejszenie objętości plików graficznych przechowywanych w pamięci mikrokontrolera PIC. Kompresję obrazów uzyskano dzięki dopasowaniu kodowania poszczególnych pikseli do możliwości wyświetlacza. Powstał także skrypt konwertujący ciągi znaków w kodowaniu UTF-8 do tablic bajtów, co ułatwia ich implementację w oprogramowaniu.

Wraz z pojawianiem się nowych zastosowań wyświetlacza OLED obsługiwanego procesorem PIC będzie możliwe dalsze rozwijanie funkcjonalności biblioteki. Wraz z doprecyzowaniem wyglądu i działania interfejsu graficznego, rozwijana będzie funkcjonalność biblioteki wspomagająca tworzenie tego interfejsu.

Planowane jest wykorzystanie pamięci w postaci karty SD w innych urządzeniach bazujących na podobnych komponentach do Modułu Inteligentnego Sterownika MIS-1. Takie rozszerzenie pamięci pozwoli na częściowe lub całkowite przeniesienie biblioteki graficznej

na dodatkową pamięć urządzenia. Największe korzyści da przeniesienie plików graficznych, które stanowią obecnie dużą część biblioteki.

Literatura

- [1] K. Stankiewicz, D. Jasiulek, J. Jagoda, J. Jura: Rozproszone systemy sterowania maszyn i urządzeń górniczych. *Maszyny Górnicze* 2016 nr 3 (147). s 54-66
- [2] D. Jasiulek, S. Bartoszek, J. Jagoda, J. Jura, Ł. Krzak: Możliwości funkcjonalne systemu rozproszonego sterowania KOGASTER. *Mechanizacja, Automatyzacja i Robotyzacja w Górnictwie*. Monografia. Praca zbiorowa. Redakcja naukowa prof. dr hab. inż. Krzysztof Krauze. Łędziny, Kraków 2015
- [3] J. Jura, S. Bartoszek, J. Jagoda, D. Jasiulek, K. Stankiewicz, Ł. Krzak: Innowacyjny system sterowania KOGASTER. *Napędy i Sterowanie* 2014. nr 7/8
- [4] J. Jura, S. Bartoszek, J. Jagoda, D. Jasiulek, K. Stankiewicz, Ł. Krzak: System sterowania KOGASTER – nowe trendy w budowie maszyn górniczych. *Maszyny Górnicze* 2014 nr 2(138). s. 43-49
- [5] A. Kozieł, D. Jasiulek, K. Stankiewicz, S. Bartoszek: Inteligentne systemy mechatroniczne w maszynach górniczych. *Napędy i Sterowanie* 2012 Nr 2 (154). s. 112-116
- [6] M. Jura: Prace koncepcyjne i oprogramowanie w zakresie aplikacji do zarządzania wyświetlaczami OLED powiązanych z typoszeregiem procesorów PIC (praca niepublikowana)
- [7] K. Stankiewicz: A method for the self-organization of a sensor network in belt conveyor exploitation. *Problemy Eksploatacji – Maintenance Problems* 2016 nr 3 (102) pp. 145-254
- [8] K. Stankiewicz: Koncepcja środowiska symulacyjnego do oceny samoorganizacji trasowania w sieci sensorycznej. *Maszyny Górnicze* 2015 nr 2. s 3-8
- [9] S. Bartoszek: Pozycjonowanie kombajnu chodnikowego w wyrobisku korytarzowym. *Maszyny Górnicze* 2016 nr 1 s. 22-35
- [10] S. Bartoszek, G. Kost: Badania i rozwój systemu pozycjonowania mobilnych maszyn górniczych KOMAG. W: Materiały na konferencję: EMTECH 2014 "Zasilanie, telemetria i automatyka w przemyśle wydobywczym. Innowacyjność i bezpieczeństwo", Szczyrk, 14-16 maja 2014 s. 166-177, ISBN 978-83-63674-10-6
- [11] M. Kostka, Ł. Krzak, A. Gawliński, D. Jasiulek, M. Latos, J. Rogala-Rojek, K. Stankiewicz, S. Bartoszek, S. Jendrysik, J. Jura: Systemy monitoringu, diagnostyki i sterowania maszyn górniczych. *Maszyny Górnicze* 2015 nr 3 s. 88-96
- [12] D. Jasiulek, J. Świder: "Mechatronic systems in mining roadheaders - examples of solutions" *Pomiary Automatyka Robotyka* 2013 nr 1
- [13] D. Jasiulek, K. Stankiewicz, M. Woszczyński: Intelligent self-powered sensors in the state-of-the-art control systems of mining machines. *Archives of Mining Science* Vol 61, No 4 (2016)
- [14] D. Jasiulek: Testing the piezoelectric energy harvester's deflection on the amount of generated energy. Springer post-conference book, entitled: "Mechatronics: Ideas, challenges, solutions and applications" Editors: J. Awrejcewicz, K. J. Kaliński, M. Kaliczyńska, R. Szewczyk. 2016. s. 95-112

- [15] D. Jasiulek: The use of lost energy for supplying the dispersed networks of sensors. *Problemy Eksploatacji - Maintenance Problems* 2015 nr 2 (97) pp. 51-59
- [16] D. Jasiulek: Propozycje zastosowania czujników samozasilających się w przemyśle wydobywczym. *Przegląd Górniczy* 2014 nr 1
- [17] S. Jendrysik, D. Jasiulek, K. Stankiewicz: System sterowania ścianowym przenośnikiem zgrzeblowym. *Maszyny Górnicze* 2015 nr 1 (141). s. 29-32

Czy wiesz, że

...w Hanowerze, w dniach 24-28 kwietnia br. odbywały się targi Hannover Messe 2017, największe i najważniejsze targi przemysłowe na świecie. W tym roku była to już 70. edycja tego wydarzenia. Dla Polski tegoroczna edycja była szczególna, ponieważ nasz kraj debiutował w roli kraju partnerskiego Hannover Messe. Mieliliśmy więc niepowtarzalną okazję do zaprezentowania osiągnięć i możliwości polskiej gospodarki, polskich firm, a także polskiej nauki. Tegoroczne targi odbywały się pod hasłem "Integrated Industry - Creating Value". Koncepcja zintegrowanej produkcji, wypromowana pod hasłem Przemysłu 4.0, jest obecna na targach w Hanowerze od kilku lat. W tym roku zwiedzający, których było aż 225 tys. (w roku 2015 było ich 217 tys.), mogli zapoznać się z możliwościami oferowanymi przez inteligentne roboty współpracujące (coboty), rozwiązania adaptacyjne oraz zintegrowane systemy energetyczne. Spośród 225 tys. uczestników pokazu, ponad 75 tys. osób przybyło z zagranicy. Jest to najlepszy wynik w 70-letniej historii Hannover Messe. Największa liczba zagranicznych gości pochodziła z Chin (9 tys.), następnie z Holandii (6,2 tys.), Indii (5,3 tys.) i z Polski – 5 tys. odwiedzających, co ustanowiło nowy rekord kraju partnerskiego.

Przegląd Mechaniczny 2017 nr 5 s.14-16