



Skanowanie kodów w automacie przemysłowej

Kody graficzne w elektronice i automatyce

Od kilku lat, na łamach Elektroniki Praktycznej, pokazujemy nowe możliwości płynące z zastosowania coraz bardziej popularnych znaczników RFID. Bezprzewodowa identyfikacja pozwala na zautomatyzowanie wielu procesów oraz ich usprawnienie zmniejszając przy tym wpływ „czynnika ludzkiego” na liczbę błędów w logistyce. Warto jednak pamiętać o technice, którą RFID często zastępują, a która choć starsza, ma wiele zalet i wciąż jest powszechnie stosowana. Mowa o kodach graficznych, a w tym kreskowych, których użytkowanie, przy obecnych możliwościach błyskawicznego przetwarzania obrazów znajduje większy sens niż kiedykolwiek dotąd.

Wbrew pozorom, zapowiadane od wielu lat zastąpienie kodów kreskowych przez znaczniki RFID wcale nie następuje. Wręcz przeciwnie – wykorzystanie kodów graficznych zdecydowanie wzrosło, a identyfikatory

radiowe użytkowane są na razie przede wszystkim w tych zastosowaniach, w których kody kreskowe by się zwyczajnie nie sprawdzały. Rozbudowanie urządzenia elektronicznego o sensor obrazu wraz



z obiektywem i ewentualnym oświetleniem jest obecnie łatwe i niedrogi, a do tego wydruk kodu kreskowego czy dwuwymiarowego jest wielokrotnie tańszy, niż stworzenie i zaprogramowanie znacznika RFID, co doprowadziło do popularyzacji kodów graficznych.

Rodzaje kodów graficznych

Kody kreskowe kojarzą się przede wszystkim z nadrukami na produktach kupowanych w sklepach, które zrewolucjonizowały handel detaliczny. Drugą z popularnych odmian są kody graficzne dwuwymiarowe, które



ELEKTRONIKA PRAKTYCZNA

często używane są do przekazywania linków internetowych do urządzeń mobilnych. I choć te pierwsze wydają się ograniczone, a te drugie – skomplikowane w odczycie, w rzeczywistości rodzajów stosowanych kodów graficznych, a nawet tych naprawdę popularnych, jest znacznie więcej. Co więcej, dzięki gotowym narzędziom, zarówno sprzętowym, jak i programowym, korzystanie z kodów graficznych można dosyć łatwo zaimplementować we własnych projektach.

Z wyglądu kody kreskowe, czy też paskowe (barcode) różnią się od kodów 2D (matrycowych) tym, że zawierają jedynie paski ułożone w linii obok siebie – w praktyce, w zdecydowanej większości są po prostu jednowymiarowe, a ich wysokość nie ma znaczenia, o ile tylko jest na tyle duża, by mogły zostać odczytane przez skaner. Klasyczne skanery odczytują te kody przejeżdżając po nich wiązką światła laserowego. Kody 2D złożone są natomiast z punktów (najczęściej prostokątnych), ułożonych w dwuwymiarowej przestrzeni, tworząc tym samym obrazek – swoistą, monochromatyczną, choć niekoniecznie, mozaikę, której najmniejsze elementy muszą być odpowiednio duże, by skaner mógł je odczytać. W efekcie kody 2D niemal zawsze pozwalają na uzyskanie większego upakowania informacji.

Naturalnie sprawia to, że układ optyczny i algorytm potrzebny do odczytu kodu dwuwymiarowego muszą być bardziej zaawansowane niż w przypadku kodów paskowych.



Dane zapisywane w kodach graficznych zazwyczaj mają postać tzw. słów. Słowa są albo grupami bitów albo konkretnymi wzorami, przy czym położenie słowa w kodzie może zmieniać graficzny wygląd takiego słowa. Używanie pojedynczych bitów, niegrupowanych w słowa jest bardzo rzadko stosowane, gdyż sensowna korekcja błędów jest możliwa dopiero wtedy, jeśli bity się ze sobą pogrupuje, bądź z góry ustali dopuszczalne kombinacje wzorów ułożenia grup bitów. Ponieważ jakość wydrukowanych

kodów może być różna, a dokładność skanera ograniczona, stosowanie korekcji błędów jest w większości przypadków koniecznością. Ma to znaczenie szczególnie w przypadku kodów dwuwymiarowych, w których nawet mały pyłek może przysłonić bit reprezentowany przez pojedynczy modul. W tych przypadkach, podczas tworzenia kodu określa się zazwyczaj albo stopień korekcji, albo odsetek słów kodowych, które mają zostać poświęcone na bity korekcji.

Ilość informacji, jaką można zapisać w kodzie zależy przede wszystkim od wybranego formatu. Kody dwuwymiarowe mają większą pojemność niż jednowymiarowe, przy czym długość jednowymiarowych nie zawsze jest z góry określona i w praktyce jest ograniczona zdolnościami odczytu skanera. W wielu przypadkach, a szczególnie w kodach 2D możliwe jest łączenie ze sobą kilku czy nawet kilkudziesięciu kodów, co pozwala zwiększyć dopuszczalną ilość przechowywanych informacji.

Ilość treści, którą można zapisać w kodzie praktycznie zawsze zależy także od rodzaju tej treści. Wynika to z faktu, że większość uniwersalnych kodów może być używana w różnych trybach. Najczęściej spotykane to tryb cyfr, alfanumeryczny i binarny. Naturalnie, do zapisu cyfr dziesiętnych potrzeba znacznie mniej bitów, niż do znaków kompletnego alfabetu, dlatego wybór trybu zapisu cyfr zazwyczaj powoduje grupowanie ze sobą kolejnych znaków, tak by np. na każdych 10 bitach zapisywać trzy cyfry, czy np. by na 6 słowach kodowych mieścić 9 cyfr. Konkretnie wartości zależą od wybranego formatu kodu i wynikają z konieczności stosowania korekcji błędów oraz z używania dodatkowych znaków funkcyjnych, umożliwiających przełączanie trybu kodu w jego środku. Mechanizm ten prowadzi do możliwości zapisu długich ciągów cyfr w trybie cyfrowym, a długich ciągów liter w trybie alfanumerycznym w ramach

jednego kodu, ograniczając jego całkowite rozmiary.

Wiele z popularnych formatów kodów umożliwia też stosowanie trybów pełnych podstawowych zestawów znaków ASCII i zestawów rozszerzonych ASCII, umożliwiającich kodowanie narodowych alfabetów łańciskich – najczęściej zaczynając od Latin 1. W najbardziej zaawansowanych pod tym względem kodach możliwe jest stosowanie alfabetów niełańciskich, np. arabskich, greckich i rosyjskich, choć ich użycie redukuje maksymalną liczbę znaków, jakie można

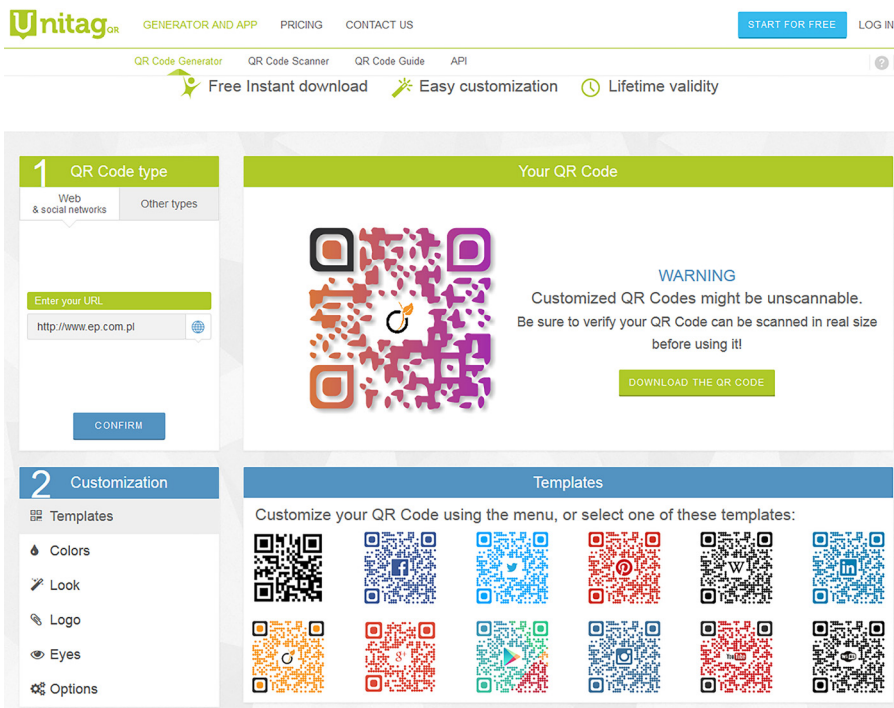
zawrzeć w pojedynczym kodzie danego formatu. Ciekawym przykładem jest kod QR, w którym jednym z podstawowych trybów jest kodowanie japońskich znaków



Kanji, co przyczyniło się do jego zdumiewającej popularności.

Naturalnie można by było po prostu przyjąć, że zera i jedynki w kodach graficznych należy oznaczać szerokością pasków lub ich obecnością (kolorem czarnym) czy też brakiem obecności (kolorem białym) i pozostawić czytelnikom swobodę tworzenia własnych rozwiązań, ale w praktyce zdecydowanie warto posłużyć się opracowanymi standardami. Dlaczego? Nie tylko dlatego, że nie ma sensu na nowo „wynajdywać koła”, tylko lepiej skorzystać ze sprawdzonych, dostępnych publicznie rozwiązań, które zostały dobrze przemyślane i udokumentowane. Przede wszystkim z tego powodu, że posługiwanie się kodami o określonym standardzie pozwala na utrzymanie kompatybilności danego rozwiązania z ogromną liczbą urządzeń i bibliotek programowych, przygotowanych do obsługi kodów graficznych. Dzięki temu implementacja obsługi kodów, czy to kreskowych, czy dwuwymiarowych, staje się nie rzadko banalnie prosta. Dotyczy to zarówno samej generacji kodów z użyciem bibliotek dostępnych dla różnych języków programowania, jak i odczytywania kodów za pomocą niedrogich skanerów, urządzeń mobilnych z kamerami, czy samodzielnie projektowanych czytników, w których również wykorzystane zostaną biblioteki przygotowane przez kogoś innego.

Ponieważ liczba opracowanych standardów jest ogromna, nie będziemy ich szczegółowo opisywać w niniejszym artykule,



a jedynie wspomnimy o najbardziej popularnych z nich, w odniesieniu do dostępnych na rynku narzędzi. Poszczególne z kodów zostały przystosowane do obsługi różnych rodzajów sytuacji oraz są mniej lub bardziej odporne na problemy typowe dla danych zastosowań.

Sposoby generowania kodów

W zależności od wybranego formatu kodu graficznego, jego generowanie będzie prostsze lub trudniejsze. Zazwyczaj warto skorzystać z gotowego mechanizmu tworzenia kodów, które zostały opracowane przez różnorodne firmy i są udostępniane bezpłatnie lub – w odniesieniu do mniej popularnych kodów – w postaci licencjonowanego oprogramowania. Mechanizmy te mają najczęściej postać albo bibliotek programowych dla różnych języków programowania, które można zaimplementować we własnych rozwiązaniach elektronicznych,

albo programów użytkowych, które po prostu uruchamia się na komputerze i w wyniku ich działania otrzymuje się gotowy plik graficzny z kodem. W tym pierwszym przypadku elektronicznie mogą wbudowywać funkcje generacji kodów do projektowanych przez siebie urządzeń lub tworzyć serwery sieciowe z generatorami, z których będą korzystały inne urządzenia. Tymczasem gotowe programy będą miały zastosowanie przede wszystkim dla logistyków i inżynierów aplikacyjnych w automatyce. Wiele generatorów jest też dostępnych bezpłatnie on-line, z których można korzystać przez przeglądarkę internetową.

Niektóre proste kody jednowymiarowe można także generować posługując się odpowiednią czcionką. Dotyczy to sytuacji, w których symbole odpowiadające poszczególnym znakom są z góry zdefiniowane i nie zależą od pozycji znaku w kodzie. Dobrym przykładem jest Code 39, przy czym np. już – dosyć podobny do niego, ale bardziej zaawansowany Code 128, choć może być wygenerowany za pomocą stosownej czcionki, wymaga ręcznego obliczenia sumy kontrolnej i dodania wynikającego z niej znaku kontrolnego na końcu ciągu.

Istotnym elementem generacji kodów graficznych jest posługiwanie się znakami funkcyjnymi i znakami ucieczki, które umożliwiają zmienianie trybów kodu, korzystanie z nietypowych alfabetów oraz włączanie funkcji specjalnych. Te ostatnie mają znaczenie w logistyce międzynarodowej, gdyż opierają się o skodyfikowany zestaw numerów aplikacji. Numery te definiują, co reprezentują następujące





po nich znaki – czy będzie to np. kod pocztowy, adres, dane wizytówki, czy numer GTIN produktu.

Dobre generatory pozwalają również nie tylko na swobodny wybór wielkości kodów i stopnia zastosowanej korekcji błędów (w ramach specyfikacji danego formatu), ale też optymalizują kod, tj. – automatycznie dobierają używane tryby i rozmiary kodów tak, by powstały symbol był jak najmniejszy. Naturalnie, jeśli kod jest generowany w postaci pliku graficznego, powinien on być zapisany w bezstratnym formacie, by nie wprowadzać zniekształceń. Najczęściej spotykanymi formatami takich plików są GIF i PNG, choć wiele bibliotek pozwala też na tworzenie JPEGów. Generator powinien także umożliwiać wybór rozdzielczości kodu i manipulowania jego docelowymi fizycznymi rozmiarami. Dzięki temu użytkownik, w zależności od tego czy wie, jakim narzędziem będzie nanoszony kod, czy też jakiej wielkości ma być gotowy nadruk, jest w stanie wygenerować optymalny plik, którego odczyt będzie później przebiegał najsprawniej.

Sposoby nanoszenia kodów

Wygenerowany kod graficzny najczęściej jest nadrukowywany na naklejkę lub bezpośrednio na docelowy obiekt. W tym pierwszym



przypadku możliwe jest użycie niemal dowolnej drukarki, choć starsze drukarki igłowe nie poradzą sobie z nowoczesnymi, gęsto upakowanymi kodami. Dawniej powstawały też kody zaprojektowane w taki sposób, by były czytelne nawet, jeśli zostaną nadrukowane za pomocą mało dokładnej drukarki igłowej, czego przykładem jest paskowy Codabar. Jeśli natomiast druk ma być wykonywany bezpośrednio na opakowaniu znakowanego produktu – a ze względu na zastosowania w logistyce, często jest to gruba, niekiedy lekko pofalowana tektura – zaleca się stosowanie kodów takich jak np. ITF-14, który świetnie się wtedy sprawdza. Trzeba też mieć na uwadze fakt, że większość kodów wymaga zachowania pewnego marginesu, zwanego też strefą cichą (quiet zone, quiet space), więc obszar zadruku musi być tak dobrany, aby zarezerwować to dodatkowe miejsce, zgodnie ze specyfikacją danego kodu. Niektóre kody dwuwymiarowe nie wymagają stosowania marginesów, a w niektórych jednowymiarowych, tak jak np. we wspomnianym ITF-14, zaleca się używanie grubej czarnej ramki naokoło pasków, celem wyraźnego zaznaczenia, że cała powierzchnia nie może być niczym zaklejana.

Rozdzielczość druku też ma znaczenie, szczególnie w przypadku kodów dwuwymiarowych.



Jest kluczowa dla MaxiCode, gdzie zlanie się ze sobą poszczególnych modułów zdecydowanie utrudnia odczyt. Powinna być więc dopasowana do rozdzielczości drukowanego obrazka, dzięki czemu nie będzie on podlegał interpolacji, co mogłoby spowodować rozmazanie modułów. Odgrywa to mniejszą rolę w kodach takich jak QR, gdzie nie ma potrzeby stosowania przerw pomiędzy modułami.

Niektóre kody, takie jak np. DataMatrix, a także Aztec Code są nanoszone bezpośrednio na różnego rodzaju produkty – płytki drukowane i metalowe lub plastikowe komponenty. Nie są wtedy nadrukowywane, ale wytrawiane, wypalane laserem lub wybijane, dzięki czemu są odporne na różne czynniki środowiskowe.

W trakcie wykonywania nadruku ważne jest, by ruch głowicy drukującej względem powierzchni zadrukowywanej był jednostajny, a przede wszystkim, by nie nastąpiło przesunięcie w innej osi niż potrzeba. W przeciwnym razie kod może zostać zniekształcony, co nawet w przypadku małych przesunięć może zupełnie uniemożliwić jego późniejszy odczyt. Ma to szczególne znaczenie w przypadku kodów dwuwymiarowych oraz kodów takich jak PDF417, a tym bardziej ich zmminiaturyzowanych wersji.

Większość kodów jest monochromatyczna, więc do ich przygotowania wystarcza czarno-biała drukarka, ale powstało też kilka formatów wielobarwnych, z tym że w praktyce nie zyskały sobie popularności. Inną sprawą jest upiększanie kodów. Od pewnego czasu, producenci towarów sprzedawanych detalicznie tworzą z kodów kreskowych EAN-13 swoiste obrazki, zachowując kluczowy fragment kodu i modyfikując jego górną część. Choć niesie to zwiększone ryzyko błędnego odczytu, w praktyce nie sprawia



problemu. Znacznie więcej pomysłowości wykazują twórcy generatorów kodów QR, którzy dopuszczają umieszczanie ikonki w środku kodu lub zamiast modułów kwadratowych, stosują np. gwiazdki. Choć kody te są niezgodne z założeniami, z jakimi powinny być tworzone, przez czytniki rozpoznawane są poprawnie, tj. jako uszkodzone ale w stopniu umożliwiającym odczyt treści. Oczywiście, gdyby taki „nadwyreżony” kod uległ dodatkowo uszkodzeniu, jego odczytanie zapewne przestałoby być możliwe.

Warto też wspomnieć o dopuszczalnych rozmiarach kodów. Większość standardów określa minimalny rozmiar modułu oraz obejmuje ograniczenia co do wysokości i maksymalnej długości (szerokości) kodu. Dotyczy to przede wszystkim aplikacji, w których zastosowanie kodu jest ściśle określone, tak jak np. w kodach EAN-13. Dzięki temu nabywcy skanerów mają pewność, że będą w stanie odczytać każdy kod obsługiwanego typu. Inne kody, takie jak np. Code 39, nie określają dokładnie maksymalnej długości. Ważne, by używany skaner był w stanie objąć i czytać cały symbol, a to zależy od mocy stosowanego lasera oraz czułości sensora. Najmniej ograniczone rozmiary mają kody takie jak DataMatrix, które można stosować np. do oznaczania układów scalonych, w związku z czym ich minimalny rozmiar nie jest ograniczony. Osoba niszcząca kod musi po prostu upewnić się, że będzie się dało go odczytać tam, gdzie będzie to potrzebne.

Sposoby odczytu kodów

Skanowanie kodu może być realizowane ręcznie lub w sposób zautomatyzowany. Ma to duże znaczenie, gdyż nie każdy kod może być czytany w sposób automatyczny. Podstawowa kwestia dotyczy możliwości lokalizacji kodu, utrzymania jego odpowiedniej orientacji względem skanera oraz tego, czy kod da się odczytać od tyłu, uzyskując inną treść, niż czytając od przodu.

Większość standardów kodów jest zaprojektowana tak, by nie można było ich przypadkiem błędnie odczytać od tyłu. W tym celu stosowane są np. słowa kodowe, które układają się zawsze poprawnie tylko w jedną stronę, a w przeciwną nie są w ogóle rozpoznawane. W przypadku kodów dwuwymiarowych stosowane są znaczniki umożliwiające zarówno wykrycie obecności kodu graficznego, jak i rozpoznanie jego orientacji w przestrzeni.

Oznacza to, że zdecydowana większość kodów graficznych może być odczytywana w sposób zautomatyzowany, choć niektóre

nie są do tego polecane. Dotyczy to przede wszystkim zminiaturyzowanych wersji większych kodów, które wymagają odczytu liniowego i istnieje obawa, że w systemach automatycznych nie zdążą być wykryte przez wirujący skaner.

Same skanery można podzielić na dwa główne rodzaje: liniowe i czytniki obrazu. Te pierwsze występują najczęściej w dwóch wersjach: ręcznej, wymagającej by operator nakierował wiązkę lasera bezpośrednio na skanowany kod, ustawiając ją w odpowiedni sposób, oraz w wersji stacjonarnej, gdzie wiązkę lasera jest rzucana na skanowaną powierzchnię pod różnymi kątami za pomocą wirującego lustra. W czytnikach obrazu wykonywane jest zdjęcie, które następnie jest analizowane przez stosowny algorytm. Naturalnie skanery kodów 2D wymagają przez to znacznie bardziej skomplikowanego oprogramowania, by były w stanie odczytać kod. Wynika to też z konieczności bardziej złożonego dekodowania słów kodowych oraz słów kontrolnych, które nierzadko obliczane są za pomocą algorytmu Reeda-Solomona, zamiast po prostu poprzez sumowanie modułu, jak to najczęściej bywa w kodach jednowymiarowych.

W przypadku dostępnych na rynku skanerów, odczytany kod jest najczęściej przekazywany poprzez wybrany interfejs do komputera. W praktyce powszechnie stosowane jest USB w trybie HID (Human Interface Device), dzięki czemu czytnik symuluje klawiaturę. To pozwala bezpośrednio wprowadzać zeskanowane kody w pola tekstowe we wszelkiego rodzaju formularzach, a użytkowanie takich skanerów nie wymaga instalowania dodatkowych sterowników. Naturalnie istnieją też bardziej specjalistyczne urządzenia, które np. prezentują zeskanowaną treść na wbudowanych wyświetlaczach, przesyłają je do sieci lub automatycznie pobierają z Internetu opis zeskanowanego przedmiotu. Jako skanery mogą posłużyć także urządzenia mobilne z aparatami fotograficznymi, które w zależności od zainstalowanej aplikacji, rozpoznają różne standardy kodów.

Do najbardziej rozpowszechnionych, a więc takich do których najłatwiej znaleźć uniwersalne skanery należą kody paskowe Code 39, Code 93, Code 128, UPC-A, EAN-13 i ITF-14. Spośród kodów 2D największą popularnością cieszą się QR Code, Aztec Code



i DataMatrix oraz – ciekawy, bo uznawany za zestawienie wielu kodów liniowych, kod PDF417.

Zastosowania i podsumowanie

Jeśli rozejrzemy się wokół siebie, okaże się że kody graficzne stosowane są częściej, niż nam się wydaje. Znajdują się nie tylko na produktach kupowanych w sklepach, ale też na ich opakowaniach zbiorczych. Można je znaleźć na receptach, kuponach zniżkowych oraz przesyłkach kurierskich a nawet na listach (to te różowe, ułożone obok siebie kreski o różnej wysokości). Są na kartach pokładowych, potrzebnych do wejścia do samolotu, na różnych biletach kupowanych przez Internet, w dokumentach rejestracyjnych pojazdów, w dokumentach osobistych w niektórych państwach, a nawet na okładce Elektroniki Praktycznej, którą właśnie trzymacie w ręce. A nawet jeśli czytacie wydanie elektroniczne, to bardzo możliwe że kody graficzne są na komponentach używanego komputera, jeśli nie na jego obudowie. Korzystacie z telefonu? Prawdopodobnie jego numer IMEI lub seryjny jest nadrukowany w postaci kodu kreskowego, a na baterii do telefonu znajduje się kod DataMatrix.

Ostatnimi czasy znacznie wzrosła popularność kodów dwuwymiarowych QR, za których pomocą przekazywane są linki do różnych serwisów internetowych (tak, jak na łamach EP), wizytówki z kontaktami czy inne dane. Statystyki wskazują, że Polacy są pod tym względem wciąż mało świadomi użyteczności kodów QR, szczególnie w porównaniu do Azjatów. Ciekawym przykładem zastosowania kodów są tagi stosowane ostatnio przez Snapchat, odsyłające młodzież do konkretnych kanałów Snapchata.

Niewątpliwie kody graficzne jeszcze przed długi czas nie znikną z naszego otoczenia. Podstawowe zmiany, jakie zachodzą w ich dziedzinie, to coraz większa popularyzacja kodów dwuwymiarowych, które na tej samej przestrzeni są w stanie zapisać więcej informacji niż kody jednowymiarowe. Powstają nawet regulacje prawne, wymuszające stosowanie konkretnych rodzajów kodów graficznych w określonych zastosowaniach, czego przykładem jest wymóg używania kodów DataBar Expanded na kuponach zniżkowych używanych w USA. Warto więc pomyśleć, czy technologia ta nie znajdzie zastosowania w aplikacjach projektowanych przez czytelników Elektroniki Praktycznej. Łatwość jej implementacji oraz wygoda posługiwania się kodami sprawia, że niejedyn projekt jest w stanie znacząco na nich skorzystać.

Marcin Karbowniczek, EP