

doi:10.15199/48.2021.12.43

## Przykłady wykorzystania wybranych cyfrowych platform do sterowania modelami obiektów

**Streszczenie.** W artykule przedstawiono krótki przegląd wybranych cyfrowych platform dostępnych obecnie na rynku elektroniki. Zostały także zaprezentowane wybrane modele obiektów zbudowanych w Katedrze Automatyki Okrętowej Uniwersytetu Morskiego w Gdyni, do sterowania których wykorzystano cyfrowe platformy, tj. model pojazdu typu line follower oraz model parkingu aut osobowych.

**Abstract.** The digital platforms have proven very useful tools for implementing projects at different levels of complexity, ranging from research to education and engineering applications. The article presents a brief overview of selected digital platforms currently available on the electronics area. Selected models of objects built at the Department of Ship Automation at the Gdynia Maritime University were also presented, i.e. the line follower vehicle model and the passenger car parking model. Digital platforms were used for control the models. (Examples of the use of selected digital platforms for control object models).

**Słowa kluczowe:** cyfrowe platformy: Arduino, Raspberry Pi, DE\_2, Edison, cyfrowe sterowanie, elektronika, programowanie, edukacja.

**Keywords:** digital platforms: Arduino, Raspberry, DE\_2, Edison, digital controlling, electronics, computer programming, education.

### Wstęp

W dzisiejszych czasach trudno wyobrazić sobie życie bez komputerów, smartfonów, tabletów, Internetu. Cyfrowe sterowanie, elektronika jest obecna prawie na każdym kroku, w każdym miejscu. Przemysł, telekomunikacja, medycyna, gospodarstwa domowe, nauka bez elektroniki już nie może funkcjonować. Podczas projektowania cyfrowego sterowania ważnym etapem jest odpowiedni dobór platformy cyfrowej. Odpowiada ona za funkcjonowanie całego systemu sterowania, stanowi element składowy, w którym zostanie zaimplementowany algorytm odpowiedzialny za pracę obiektu sterowania. Najczęściej wykorzystany do sterowania mikrokontroler powinien posiadać wydajny procesor, dużą ilość pamięci RAM, interfejs komunikacji bezprzewodowej z komputerem, odpowiednią ilość portów wejścia/wyjścia, umożliwiającą połączenie z np. czujnikami, silnikami, kamerą itp.

Obecnie na rynku dostępnych jest wiele cyfrowych platform, które można wykorzystać do sterowania różnymi obiektami, przykładowo Arduino, Raspberry Pi, DE\_2, Edison. Są one funkcjonalne, posiadają dodatkowe moduły, które umożliwiają rozbudowę. To spowodowało, że cyfrowe platformy stały się w ostatnim okresie bardzo popularne [1 - 5]. Są one wykorzystywane nie tylko w przemyśle, medycynie, ale także na różnym poziomie kształcenia i szkolenia. Wielu pasjonatów programowania, elektroniki i techniki cyfrowej buduje własne roboty, obiekty sterowania. Przykłady wykorzystania różnych platform do sterowania obiektami zostały przedstawione w wielu pracach [1, 6-13]. Ponadto w pracy [14] zostały przedstawione przykłady zastosowania platformy Arduino do nauki programowania w języku C/C++, nauki mechatroniki, robotyki, elektroniki, tworzenia interfejsów graficznych, sterowania robotem. Arduino oraz Raspberry są wykorzystywane w nauczaniu sterowania obiektem pneumatycznym [13]. W pracy [11] opisano gramerkę laserową CNC, do sterowania której wykorzystano platformy Raspberry oraz DE\_2. W artykule [15] omówiono zastosowanie Raspberya do nauki programowania w języku Python. Przedstawiono w nim także ciekawe porównanie wyników nauczania programowania w języku C, Matlab oraz Python. Z przedstawionej analizy wynika, że język Python zyskuje coraz większą popularność w różnych zastosowaniach przemysłowych. Z analizy dostępnej literatury wynika także, że do sterowania obiektami oraz w procesie kształcenia często jest wykorzystywana platforma DE\_2. W licznych pracach autorki zostały przedstawione przykłady

wykorzystania DE\_2 do sterowania różnymi modelami, np. modelami różnych pojazdów, modelem sejfu, elektrowni okrętowej, inteligentnego domu, modelem urządzenia do śledzenia słońca i ładowania akumulatora [6-9]. Inny przykład wykorzystania platformy DE\_2 zaprezentowano w pracy [10], w którym omówiono moduł badawczy oparty na magistrali SPI (ang. Serial Peripheral Interface) z mikrokontrolerem pracującym jako urządzeniem Master oraz urządzeniami Slave podłączonymi do magistrali. Ponadto w pracy [12] przedstawiono analizę porównawczą dla protokołu komunikacyjnego Modbus RTU z wykorzystaniem sterownika PLC i układu programowalnego FPGA będącego na wyposażeniu platformy DE\_2. Znaczna ilość już omówionych zastosowań platformy DE\_2 spowodowała pominięcie w prezentowanym artykule opisu kolejnych przykładów.

W artykule przedstawiono przegląd cyfrowych platform dostępnych obecnie na rynku elektroniki. Zostały także zaprezentowane wybrane modele obiektów zbudowanych w Katedrze Automatyki Okrętowej (KAO) Uniwersytetu Morskiego w Gdyni (UMG), tj. model pojazdu typu line follower oraz model parkingu aut osobowych. Do sterowania modeli zostały wykorzystane cyfrowe platformy.

### Przegląd cyfrowych platform

#### Platforma Arduino

Jedną z częściej wykorzystywanych cyfrowych platform uruchomieniowych jest produkt firmy Arduino. Obecnie firma Arduino oferuje następujące modele: Uno, Leonardo, Mega, Nano i Portenta H7. Spośród wymienionych platform często jest wykorzystywana wersja Mega 2560 Rev3. Każda platforma Arduino wyposażona jest w 8-bitowy procesor z rodziny Atmel AVR, o maksymalnej częstotliwości zegara 16 MHz. Do komunikacji z komputerem jest wykorzystywany port szeregowy UART (ang. Universal Asynchronous Receiver-Transmitter). Ważną cechą platformy Arduino jest możliwość jej rozbudowania o dodatkowe elementy. Przykładowo dostępne są moduły umożliwiające sterowanie silnikami, komunikację poprzez WiFi, Bluetooth oraz na podłączenie klawiatury lub wyświetlacza [16]. Mikrokontroler można programować przy użyciu portu szeregowego UART. Firma Arduino udostępnia własne środowisko programistyczne Arduino IDE (ang. Integrated Development Environment), które można pobrać ze strony producenta [17]. Arduino IDE jest wyposażone w bibliotekę C/C++ i tzw. projekt Wiring, który upraszcza podstawowe operacje wej-

ścia / wyjścia. Do programowania mikrokontrolera niezbędne jest połączenie z komputerem za pośrednictwem łącza USB.

Najnowszą wersją Arduino jest platforma Portenta H7, która została zaprezentowana w styczniu 2020 roku na targach CES (ang. Consumer Electronics Show) [18]. Została ona przygotowana przede wszystkim dla przemysłu IoT (ang. Internet of Things) i sztucznej inteligencji. Platforma Portenta posiada dwurdzeniowy mikrokontroler STM32 H747, który może wykonywać równolegle różne zadania. Na płytce znajduje się także kryptograficzny koprocesor NXP SE050C2, ładowarka do ogniw litowo-polimerowych, nowoczesne złącze USB typu C, a także moduły WiFi i Bluetooth. Jest ona kompatybilna ze starszymi modułami.

#### Platforma Raspberry Pi

Dotychczas na rynku pojawiły się cztery podstawowe wersje platformy Raspberry Pi, najnowszą wersją jest Raspberry Pi 3B+. Płytkę umożliwia komunikację bezprzewodową, za którą odpowiedzialny jest chip Cypress CYW43455 obsługujący WiFi oraz Bluetooth. Platforma posiada czterordzeniowy procesor Broadcom BCM2837B0 o taktowaniu zegara 1,4 GHz, 1 GB pamięci RAM oraz slot na kartę micro SD [19]. Dodatkowo płytka wyposażona jest w 4 porty USB i port LAN. Do przesyłania obrazu i dźwięku zastosowano standardowe porty, tj. HDMI (ang. High Definition Multimedia Interface). Na płytce znajduje się 40 portów wejścia/wyjścia. Platforma Raspberry Pi, jako mikrokomputer, wyposażona jest we własny system operacyjny, przy czym podstawowym jest system Raspbian. Płytkę umożliwia programowanie w języku C lub C++, Javie i Python. Raspberry może również zostać wykorzystany jako serwer internetowy, można go programować w języku HTML, PHP oraz w NET Core.

#### Platforma Edison

Edison firmy Intel jest kolejną dostępną platformą, której pojawienie się wynikało z powstania Internetu Rzeczy [20]. Platforma posiada dużą moc obliczeniową, jednak jej podstawowym zadaniem jest zbieranie danych, przetwarzanie informacji i przesyłanie ich do chmury obliczeniowej, gdzie wykonywane są różne skomplikowane operacje. Na platformie Edison znajduje się dwurdzeniowy procesor Intel Atom wykonany w technologii 22 nm, który pracuje z częstotliwością 500 MHz. Dodatkowo płytka posiada koprocesor Intel Quark o częstotliwości pracy 100 MHz, 4 GB pamięci flash oraz 1 GB zintegrowanej pamięci RAM [21]. Za komunikację bezprzewodową odpowiedzialne są wbudowane moduły WiFi oraz Bluetooth. Platforma posiada również własny system operacyjny. Możliwe jest tworzenie programów za pomocą trzech kompilatorów: Intel XDK IoT Edition, Arduino IDE oraz Eclipse. Algorytmy sterowania można przygotować w języku C oraz C++, Python oraz JavaScript.

#### Platforma DE-2

Kolejną dostępną na rynku platformą jest moduł edukacyjny Altera DE\_2-115, której producentem jest firma Terasic Technologies [22]. Platforma DE\_2 jest często wykorzystywana do nauki obsługi oraz programowania układów FPGA (ang. Field Programmable Gate Array) oraz CPLD (ang. Complex Programmable Logic Devices). Działanie tych układów określa zaimplementowane oprogramowanie. Prosta obsługa DE\_2, uniwersalność oraz duża ilość portów rozszerzeń pozwala na realizację zarówno prostych układów logicznych oraz bardziej skomplikowanych projektów.

Płytkę DE\_2-115 wyposażona jest w procesor EP4CE115 z rodziny Cyclone IV firmy Altera, 2 MB pamięci SRAM, 128 MB SDRAM, 8 MB pamięci Flash oraz czytnik

kart SD. Do komunikacji zewnętrznej płytka posiada port USB oraz port Ethernet. Dostępne są również porty do nadawania formatów audio oraz wideo. Płytkę posiada także gniazdo 40 pinów wejścia/wyjścia. Uruchomienie płytki wymaga połączenia, poprzez kabel USB, z komputerem, na którym powinny być zainstalowane sterowniki USB Blaster. Do zestawu DE\_2 dołączony jest panel sterowania oraz środowisko programistyczne Quartus, które pozwala na obsługę układów programowalnych. Platforma DE\_2 umożliwia tworzenie różnych projektów przy pomocy języka programowania sprzętu VHDL lub Verilog, kompilowanie, debugowanie projektów oraz symulacje czasowe zaprojektowanych układów.

#### Autonomiczny pojazd rozpoznający znaki drogowe i poruszający się po wyznaczonej trasie

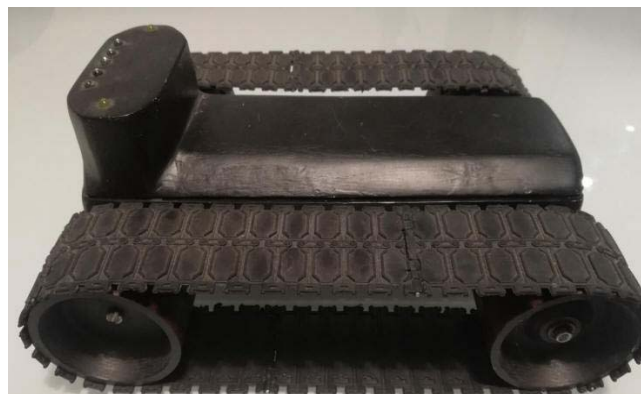
Widocznym znakiem realizowanej obecnie rewolucji technologicznej są w pełni zautomatyzowane samochody. Branża motoryzacyjna inwestuje w rozwój znaczne środki finansowe. Dąży do znacznego wyeliminowania z transportu drogowego człowieka. Wdrożenie całkowitej automatyzacji w transporcie drogowym wymaga jednak czasu niezbędnego na badania.

#### Budowa pojazdu

Zaprojektowany i wykonany w KAO, w ramach pracy inżynierskiej [23], autonomiczny pojazd typu line follower (rys. 1) realizuje następujące zadania:

- pokonuje drogę po zamodelowanej trasie,
- wykrywa oraz odpowiednio reaguje na znaki drogowe.

Autonomiczność pojazdu ogranicza ingerencję człowieka jedynie do uruchomienia lub awaryjnego zatrzymania pojazdu. Za sterowanie ruchem pojazdu odpowiada zaimplementowany algorytm, który został opracowany w środowisku Visual Studio Code, w języku Python, przy użyciu biblioteki OpenCV. Biblioteka ta została wykorzystana do obróbki obrazu. Zbudowany pojazd porusza się po zamodelowanej trasie, która została wytyczona na elastycznej wykładzinie. Na białym tle zostały umieszczone czarne linie, po których porusza się pojazd (rys. 2). Podczas ruchu pojazd uwzględnia także znaki drogowe. Można je umieszczać przed skrzyżowaniami. W zaimplementowanym algorytmie uwzględniono cztery znaki: nakaz jazdy w prawo, w lewo, zwracania oraz znak stop. Znaki wybrano w taki sposób, aby zobrazować reakcję pojazdu po ich wykryciu. Na karoserii pojazdu została umieszczona kamera oraz czujnik ultradźwiękowy. Pojazd posiada własny napęd gąsienicowy, który jest napędzany przez dwa silniki o przekładni ślimakowej. Posiada również własne oświetlenie LED, które ułatwia detekcję znaków.

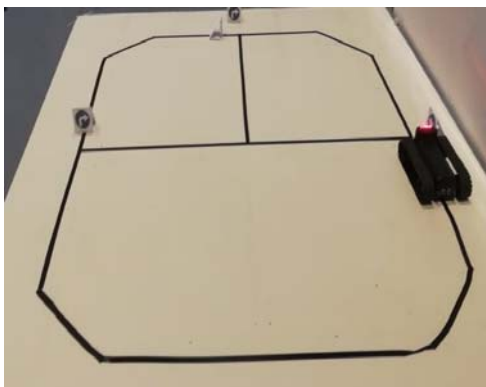


Rys. 1. Wygląd pojazdu [23]

Projekt pojazdu został wykonany, przy użyciu programu SketchUp, jako trójwymiarowy model. Wielkość i wymiary

pojazdu (24 cm długości oraz 9 cm szerokości) zostały dostosowane do wykorzystanych elementów składowych, tj. do np. baterii, silnika, mikrokontrolera, które zostały umieszczone wewnątrz pojazdu. Przy określeniu wielkości pojazdu uwzględniono także jego prędkość oraz mobilność, co spowodowało wykorzystanie silników ślimakowych i napędu gąsienicowego. Większość elementów została wykonana samodzielnie, np. wytoczono koła, wykonano z żywicy epoksydowej konstrukcję [23]. Natomiast część elektroniczną pojazdu wykonano z elementów dostępnych w sklepach modelarskich.

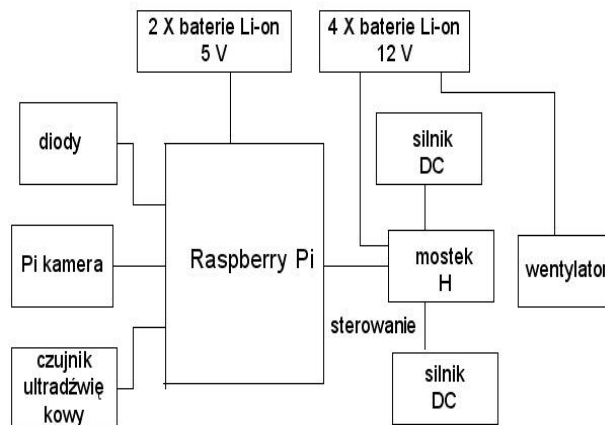
W zbudowanym pojeździe wykorzystano dwie gąsienice z kołami napędowymi umieszczonymi z tyłu pojazdu oraz kołami prowadzącymi usytuowanymi na przedniej osi. Konstrukcja ramy pojazdu została wykonana z kątowników aluminiowych o wymiarach 2 x 15 x 15 mm. Pojazd posiada dużą zdolność manewrowania. Ruchy pojazdu realizowane są poprzez odpowiednie uruchamianie prawego lub lewego silnika. Koła przednie i prowadzące gąsienice posiadają dwa łożyska, jedno zostało usytuowane od strony karoserii oraz drugie - po stronie zewnętrznej. Dzięki temu koła poruszają się stabilnie, nie stawiają dużego oporu. Koła tylne, odpowiedzialne za przeniesienie napędu na gąsienice, zostały zamontowane bezpośrednio na niezależnych wałkach przekładni ślimakowych. Jako napęd wykorzystano silniki szczotkowe prądu stałego. Maksymalna prędkość obrotowa na wale wynosi 60 obr/min. Przeprowadzone badania i obliczenia wykazały, że średnia prędkość pojazdu wynosi około 792 m/h. Na rysunku 3 przedstawiono schemat blokowy konstrukcji elektronicznej zbudowanego pojazdu.



Rys. 2. Zamodelowana trasa [23]

Głównym elementem pojazdu jest platforma Raspberry Pi, który steruje poszczególnymi podzespołami. Obsługują one sterownik silników, diody LED, czujnik ultradźwiękowy, są odpowiedzialne za zasilanie pojazdu z akumulatorów. Ponadto Raspberry wspiera port MIPI Camera Serial Interface (CSI), który umożliwia komunikację z kamerą. Na pojeździe zamontowano kamerę Raspberry Pi, która jest odpowiedzialna za detekcję drogi oraz znaków drogowych. Wykorzystany czujnik ultradźwiękowy HC-SR04 umożliwia pomiar odległości w zakresie od 2-200 cm. Dzięki informacji o odległości do przeszkody lub wykrytego znaku drogowego możliwa jest odpowiednia zmiana kierunku ruchu pojazdu. Moduł dokonuje pomiaru odległości przy pomocy fali dźwiękowej o częstotliwości 40 kHz. Do mikrokontrolera wysyłany jest sygnał, w którym odległość zależy od czasu trwania stanu wysokiego na odpowiednim porcie. Do sterowania silnikami został wykorzystany dwukanałowy sterownik silnika prądu stałego L298N, tzw. mostek H. Umożliwia on sterowanie silnikami w kilku trybach, które warunkują różne zachowanie pojazdu. Do zasilania części logicznej wykorzystano dwa akumulatory Li-ion typu 18650, natomiast do

zasilania sterownika silników 4 akumulatory tego samego typu. Na mostku H występuje sterowanie PWM, w celu uzyskania określonej prędkości pojazdu można określić wartość napięcia.



Rys. 3. Schemat blokowy konstrukcji elektronicznej pojazdu

Podczas prac uruchomieniowych oraz testowania pojazdu zaobserwowano podwyższenie temperatury mikrokontrolera oraz sterownika silników. Spowodowało to konieczność zastosowania dodatkowych elementów wspomagających odprowadzanie ciepła, tj. na płytce Raspberry zamontowano dwa radiatory. Mimo że sterownik silników posiada fabryczny radiator to jednak w czasie ciągłej eksploatacji pojazdu okazał się on niewystarczający. Dlatego dodatkowo w tylnej części podwozia umieszczono wentylator, który wciąga powietrze z otoczenia pojazdu do jego środka. Dzięki tym dodatkowym elementom wyeliminowano problemy związane z przegrzewaniem się podzespołów.

Obraz z kamery przyjmuje nieco inną postać w świetle dziennym oraz sztucznym. Z przeprowadzonych badań wynika, że istotny wpływ na detekcję miały również cienie rzucane przez elementy otoczenia. W celu wyeliminowania problemów z oświetleniem, na frontowej części pojazdu zostały zamontowane cztery białe diody LED. Świecenie diod LED następuje po starcie mikrokontrolera. Po przeprowadzeniu testów na pojeździe dodatkowo zamontowano kolorowe diody LED, które prezentują działanie algorytmu po wykryciu znaków drogowych. Dla znaków nakazu jazdy w prawo lub w lewo migają diody żółte, które symbolizują kierunkowskazy. Natomiast dla znaku stop lub po wykryciu przeszkody, zapalają się diody czerwone, które obrazują samochodowe światło stop.

### Oprogramowanie sterujące

Zbudowany pojazd porusza się po trasie bez ingerencji człowieka. Na polskich drogach pojazdy autonomiczne nie są widokiem codziennym, chociaż latem 2019 taki autobus dowoził turystów do gdańskiego ZOO [24]. Nieco inna sytuacja występuje za oceanem, gdzie są prowadzone zaawansowane prace i testy nad pojazdami autonomicznymi [25].

Działanie algorytmu zaimplementowanego w pojeździe podzielono na trzy główne etapy:

- wyszukiwanie drogi, które odbywa się na podstawie obrazu z kamery. W celu uzyskania odpowiedniej rozdzielczości oraz barwy obraz ten podlega obróbce,
- wykrywanie znaków drogowych, można je umieszczać na skrzyżowaniach. Wykrycie znaków odbywa się przy pomocy obrazu z kamery oraz czujnika ultradźwiękowego, który umożliwia określenie odległości do znaku,
- podejmowanie decyzji i sterowanie pojazdem, decyzje są podejmowane na podstawie zaimplementowanego algorytmu, przy uwzględnieniu zebranych danych o otocze-

niu. Sterowanie pojazdem polega na przesłaniu informacji o kierunku pracy silników.

Podczas wyboru algorytmu szukania trasy początkowo były rozważane dwa rozwiązania, tj. z uwzględnieniem sieci neuronowych oraz metoda line follower. Ze względu na spore problemy zrezygnowano z zastosowania sieci neuronowych. Zdecydowanie prostszą okazała się metoda line follower, czyli podążanie za linią, która została w pojeździe ostatecznie zaimplementowana.

W procesie przetwarzania obrazu główną rolę odgrywa uzyskanie tzw. obszaru zainteresowania ROI (ang. Region of Interest). Przy czym ważne są dwa takie obszary, pierwszy to dolny widok z linią trasy. Natomiast drugi zawiera fragment od połowy obrazu w górę, gdzie najczęściej znajdują się znaki drogowe. Wyznaczony obszar jest przekształcany z obrazu kolorowego na szary. Następnie obraz jest przekazywany jako parametr wejściowy dla funkcji odpowiedzialnych za detekcję. W opracowanym sterowaniu najwięcej problemów było z implementacją algorytmu wyszukiwania drogi. Dla komputera obraz to dwuwymiarowa tablica pikseli. W dalszym kroku opracowanego algorytmu na podstawie ROI utworzona została perspektywa, czyli obraz widziany pod kątem został rzutowany na obraz widziany z góry. W efekcie końcowym każdy obraz zanim trafił do metody detekcji trasy został wstępnie przetworzony. W kolejnym kroku realizacji algorytmu detekcji trasy następuje wyodrębnienie z obszaru ROI koloru czarnego, reprezentującego drogę, po której powinien poruszać się pojazd. Następnie zwrócone zostają kontury największego obiektu, czyli drogi. Za wyodrębnienie kolorów i znalezienie konturów odpowiadają metody dostępne w bibliotece OpenCV.

Na podstawie zdefiniowanych konturów trasy wyznaczany jest wierzchołek najbardziej wysunięty do góry, od współrzędnych którego odejmowana jest wartość położenia osi przebiegającej przez środek ROI. Otrzymany wynik oznacza kąt wychylenia pomiędzy centralną osią, a kierunkiem w jakim podąża pojazd. Wartość ujemna oznacza kierunek jazdy w prawo, natomiast dodatnia - kierunek w lewo. Ponadto jeśli wartość bezwzględna wychylenia mieści się w przedziale błędów, tj. 100 pikseli, pojazd nie zmienia swojego kierunku. W przypadku, gdy trasa nie zostanie wykryta wysyłana jest komenda powodująca zatrzymanie pojazdu. W efekcie końcowym z wejściowego ROI otrzymujemy parametr określający kierunek w jakim ma poruszać się pojazd.

Dla algorytmu wykrywania znaków drogowych również początkowo rozważano dwa rozwiązania, tj. detekcję Haar-like oraz detekcję wykorzystującą kształt i kolor, którą ostatecznie wykorzystano. Detekcja Haar-like polega na wykrywaniu pożądanych obiektów na obrazie, jest powszechnie wykorzystywana przy np. wykrywaniu twarzy [26]. W drugim algorytmie odbywa się wyszukiwanie w obrazie kształtów typowych dla znaków, następnie wyodrębnienie obszarów, które pozwolą zidentyfikować znak. Algorytm odpowiedzialny za wykrywanie znaków również na wejściu otrzymuje fragment obrazu przesłanego przez kamerę. Na etapie końcowym następuje zdefiniowanie, na podstawie koloru obszarów, określonego znaku.

Algorytm sterowania pojazdem został opracowany w języku Python. O wyborze języka zadecydowała przede wszystkim duża ilość zewnętrznych bibliotek. Kod został napisany w darmowym środowisku IDE Visual Studio Code. Program po zainicjowaniu wykonywany jest w pętli tak długo, jak długo kamera przekazuje obraz lub do czasu przerwania działania programu. Zadaniem pojedynczego cyklu pętli jest określenie, na podstawie otrzymanego obrazu, informacji o kolejnym ruchu pojazdu. Algorytm zaimplementowanej logiki został przedstawiony na rysunku 4.



Rys. 4. Algorytm zaimplementowanej logiki

Podczas testowania przygotowanego oprogramowania pojawiły się problemy, które w wyniku wprowadzenia poprawek zostały wyeliminowane. Przykładowo początkowo do zamodelowania trasy wykorzystano taśmę o szerokości 5 cm, która była widoczna na prawie połowie obrazu z kamery, szerokość taśmy zmieniono na 2 cm. Problemy stwarzało także światło dzienne i sztuczne. Cień rzucany przez pojazd był interpretowany przez algorytm jako trasa do przebycia. Dodatkowe diody, symbolizujące światła samochodowe, zamontowane na froncie pojazdu zwiększyły skuteczność algorytmu wyszukiwania trasy. Niestety światło z diod odbijało się od znaków drogowych pokrytych laminatem, co utrudniało ich rozpoznanie. Wprowadzono również zmiany w kodzie programu, przykładowo odpowiednio dobranie parametrów związanych z prędkością oraz wartością odchylenia od trasy ułatwiło podjęcie decyzji o kierunku ruchu pojazdu. Jednak nie do końca rozwiązano problemy związane z oświetleniem dziennym i sztucznym. Przewiduje się dalszą rozbudowę i modyfikację zbudowanego pojazdu, np. dodanie nowych funkcjonalności, powrót do rozwiązania z wykorzystaniem sieci neuronowych, chmury obliczeniowej.

### Cyfrowe sterowanie modelem parkingu

Do sterowania modelem parkingu dla samochodów osobowych (rys. 5), zbudowanego w ramach pracy dyplomowej [27], została wykorzystana platforma Arduino Nano. Ilość dostępnych wejść i wyjść okazała się wystarczająca do obsługi czujników, serwomechanizmów, czytnika RFID (ang. Radio-Frequency Identification) oraz LCD. Zaprojektowana i zbudowana makieta parkingu jest kolejnym przykładem na wykorzystanie do sterowania platformy uruchomieniowej. Model odwzorowuje rzeczywisty parking z osobnym pojazdem i wyjazdem, automatycznie otwieranymi szlabanami oraz zliczaniem i wyświetlaniem ilości miejsc do parkowania. Przy wyborze platformy istotnym czynnikiem była możliwość rozwiązania problemu jednowątkowości Arduino. Mikrokontroler ten posiada tylko jeden rdzeń, pomimo tego obsługuje on oba szlabany jednocześnie.

Model parkingu zawiera:

- makietę o wymiarach 30x40 cm, która została ograniczona przy pomocy band,
- pięć modeli samochodów osobowych (o wymiarach 5x8 cm każde),
- dwa niezależnie działające szlabany, sterowane za pomocą serwomechanizmów,
- dwa czujniki podczerwieni wykrywające pojazd znajdujący się przy szlabanie,
- wyświetlacz LCD, usytuowany przy wjeździe, informujący o ilości wolnych miejsc,
- odbiornik RFID/NFC, zamontowany przy wyjeździe, niezbędny do obsługi karty płatniczej, symuluje on dokonanie opłaty za skorzystanie z parkingu.

Podstawę makiety parkingu tworzy korkowa płyta. Jako bariery wykorzystano kanał elektroinstalacyjny, w którym ukryto przewody. Wyświetlacz oraz platforma Arduino zostały umieszczone w puszcze elektrycznej. Moduł z odbiornikiem NFC (ang. Near Field Communication) został zamontowany w drugiej puszcze elektrycznej umieszczonej przy szlabanie wyjazdowym. Schemat blokowy modelu parkingu został przedstawiony na rysunku 6. Do opracowania schematu połączeń układów elektronicznych został wykorzystany darmowy program Fritzing 0.9.3 [28].

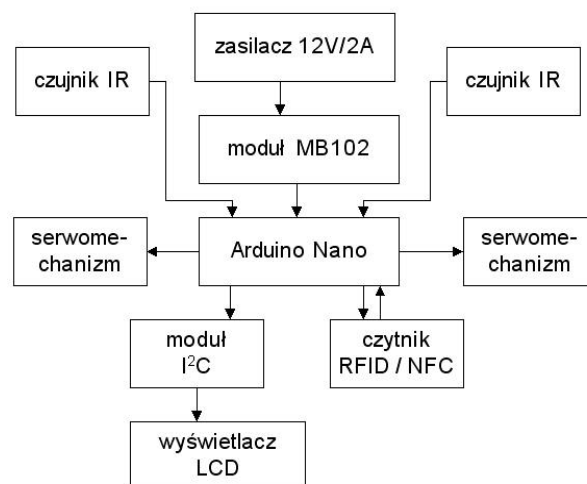


Rys. 5. Widok modelu parkingu [27]

Głównym elementem zasilającym i stabilizującym napięcie jest dodatkowo zainstalowany moduł MB102. Umożliwia on zasilanie modelu napięciem 3,3 V lub 5 V, przy maksymalnym prądzie wyjściowym 700 mA. Wybór odpowiedniego napięcia wyjściowego odbywa się za pomocą zworek, które należy odpowiednio przełączyć. Na płytce modułu zostały umieszczone dwa stabilizatory napięcia. Dodatkowo moduł został wyposażony w diodę sygnalizującą podłączenie zasilania oraz wyłącznik. Na potrzeby modelu parkingu napięcie wyjściowe zostało ustawione na 5 V.

Do obsługi szlabanów zostały wykorzystane dwa serwomechanizmy SG90, sterowanie odbywa się za pomocą sygnału PWM, którego obsługa jest dostępna z poziomu Arduino. Serwomechanizmy tworzą zamknięte układy sterowania ze sprzężeniem zwrotnym. W obudowie znajduje się silnik, przekładnia, układ sterujący oraz potencjometr, który pozwala ustalić aktualną pozycję [29]. Każdy z serwomechanizmów współpracuje z czujnikiem odbiciowym podczerwieni OKY3127, który wykrywa znajdujący się przed nim pojazd. Na płytce czujnika znajduje się potencjometr przy pomocy którego można ustawić czułość, można więc określić jak blisko powinien być pojazd. Zgodnie z kartą katalogową producenta zakres ten wynosi od 2 do 20 cm [30]. W porównaniu do czujników ultradźwiękowych, czujnik podczerwieni jest prostszy. Jednak w tym przypadku

nie jest możliwe określenie odległości od przeszkody, a jedynie czy znajduje się ona w zasięgu.



Rys. 6. Schemat blokowy konstrukcji elektronicznej modelu parkingu

Do komunikacji wyświetlacza LCD z Arduino został wykorzystany konwerter I<sup>2</sup>C (ang. Inter-Integrated Circuit), dzięki któremu zmniejszono ilość niezbędnych wejść i wyjść z 16 do 4. Do transmisji danych na wyświetlacz zostały wykorzystane dwie dwukierunkowe linie analogowe SDA (ang. Serial Data line) i SCL (ang. Serial Clock Line). Konwerter I<sup>2</sup>C jest osobnym elementem, został on przyłutowany do płytki wyświetlacza LCD.

Do symulacji płatności został wykorzystany czytnik PN532, który obsługuje protokoły RFID oraz NFC. Umożliwia on odczyt danych na częstotliwości 13,56 MHz w paśmie 2 GHz. Układ jest zasilany napięciem 5 V i komunikuje się z Arduino za pomocą standardu I<sup>2</sup>C. W rzeczywistych warunkach poprawność płatności zbliżeniowej jest weryfikowana przez terminal. W zbudowanym modelu parkingu weryfikacja nie została zaimplementowana, dlatego szlaban można otworzyć każdą kartą RFID. Jednak jest możliwość zastrzeżenia wyłącznie wybranych kart, co podczas rozbudowy modelu zostanie uwzględnione.

Oprogramowanie sterujące zostało przygotowane w języku C++, przy pomocy darmowego środowiska Visual Studio Code. Edytor ten jest znacznie bardziej rozbudowany niż standardowy edytor przeznaczony dla Arduino.

Arduino Nano posiada tylko jeden rdzeń, może więc realizować jednocześnie tylko jedną operację [5]. Problem obsługi jednoczesnego wjazdu oraz wyjazdu aut został rozwiązany programowo za pomocą systemu operacyjnego czasu rzeczywistego RTOS (ang. Real Time Operating System). Dla Arduino, za pomocą kodu, nie ma możliwości obsługi kilku czynności. Możliwe jest jednak szybkie przełączanie między zadaniami, dzięki czemu powstaje wrażenie jakby czynności były wykonywane jednocześnie. Takie przełączanie zadań, tzw. przełączanie kontekstu, zostało wykorzystane w zbudowanym modelu parkingu.

W trakcie budowy i testowania modelu pojawiło się kilka problemów technicznych. Problemem było prawidłowe umiejscowienie czujników. Odpowiednia ich wysokość ma istotne znaczenie dla poprawności wykrywania pojazdów. Dlatego konieczne było umieszczenie czujników jak najbliżej powierzchni parkingu. Dodatkowym utrudnieniem okazało się zbyt czułe działanie czujników podczerwieni. Podczas testów z modelami pojazdów okazało się, że czujnik wykrywał model na ułamek sekundy, po czym wracał do stanu pierwotnego, co powodowało bardzo szybkie otwarcie i zamknięcie szlabanu wjazdowego i zmniejszenie liczby

wolnych miejsc. W celu poprawy w algorytmie zastosowano dodatkowe opóźnienie zapewniające, że czujnik pozostawał w stanie wykrycia co najmniej przez jedną sekundę, dopiero po upływie tego czasu szlaban mógł zostać otwarty.

W przyszłości przewiduje się rozbudowę modelu parkingu, np. możliwy jest montaż głównego oświetlenia, które byłoby automatycznie włączane po wykryciu ruchu pojazdu. Przewiduje się także zastosowanie mniejszych czujników podczerwieni, umieszczenie czujników dymu, których uruchomienie spowodowałoby awaryjne otwarcie szlabanów oraz włączenie alarmu pożarowego. Kolejnym usprawnieniem może być wyświetlanie na LCD bieżącej daty i godziny, aktualnej temperatury i stanu powietrza, ciśnienia. Analiza pracy [31] podsunęła pomysł na sprawdzenie możliwości zastosowania do sterowania modelem parkingu algorytmów genetycznych oraz IoT.

## Wnioski końcowe

Różnica między dostępnymi obecnie na rynku cyfrowymi platformami spowodowana jest przede wszystkim zastosowaniem innego procesora i dostępnymi dodatkowym wyposażeniem. W zasadzie są one kompatybilne z Arduino lub Raspberry Pi, z uwzględnieniem układów zwiększających możliwości funkcjonalne. Arduino oraz DE-2 są platformami z dedykowanym środowiskiem programowania, natomiast Raspberry jest mikrokontrolerem wyposażonym w system operacyjny. Liczne modele oraz algorytmy sterowania zaprezentowane w literaturze oraz Internecie świadczą o ogromnej popularności platform uruchomieniowych. Ich wybór jest zależny od rodzaju analizowanego projektu, wiedzy i możliwości projektanta, możliwości finansowych, czasu niezbędnego na wykonanie projektu. Na uwagę zasługuje fakt, że platformy są wykorzystywane w wielu dziedzinach życia codziennego, do sterowania różnymi obiektami, zabawy, rozwijania własnych zainteresowań przez osoby w różnym wieku i o różnym poziomie wiedzy. Są one wykorzystywane w pracach badawczych oraz w procesie nauki i kształcenia. Również w KAO w UMG kontynuowane są badania z zastosowaniem różnych platform uruchomieniowych, które współpracują z różnymi obiektami. Między innymi obecnie, w ramach kolejnej pracy dyplomowej, realizowane są końcowe prace uruchomieniowe i testujące związane ze zbudowanym modelem zdalnie sterowanego pojazdu poruszającego się po schodach oraz niewielkich wzniesieniach.

**Autorzy:** dr inż. Krystyna Maria Noga, Uniwersytet Morski w Gdyni, Katedra Automatyki Okrętowej, ul. Morska 81-87, 81-225 Gdynia, E-mail: k.noga@we.umg.edu.pl.

## LITERATURA

- [1] Boxall J., Arduino 65 praktycznych projektów, Wydawnictwo Helion SA, 2014
- [2] Guttag J. V., Introduction to Computation and Programming Using Python: With Application to Understanding Data, MIT Press, 2016
- [3] www.malinowepi.pl (dostęp styczeń 2021)
- [4] Mank S., Arduino dla początkujących. Podstawy i szkice, Wydawnictwo Helion, 2014
- [5] Hoffman J., Zostań mistrzem Arduino, Wydawnictwo Helion, 2019
- [6] Noga K.M., Cyfrowe sterowanie z zastosowaniem układów programowalnych, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, Nr 40, 2014, 73-76
- [7] Noga K. M., Modele sterowane cyfrowo, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, Nr 46, 2015, 81-84
- [8] Noga K. M., M. Radwański M., Modern improvements in the digital logic laboratory, Technological Developments in Networking, Education and Automation, USA, *Springer* 2010, 109-114
- [9] Noga K. M., Wykorzystanie platform Arduino UNO oraz DE-2115 do sterowania modelami aut, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, Nr 60, 2018 r., 85-90
- [10] Noga K. M., Rabczuk D., Laboratoryjny system cyfrowy programowany przez Ethernet oparty na magistrali SPI, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, Nr. 65, 111-114, 2019 r., ISSN 2353-1290, doi: 10.32016/1.65.18
- [11] Noga K. M., Zastosowanie układów programowalnych do cyfrowego sterowania modelami obiektów, *Przegląd Elektrotechniczny*, R. 96, Nr 2/2021, 87-92, ISSN 0033-2097, doi: 10.15199/48.2021.02.21
- [12] Noga K. M., M. Rybczak, Kacperkiewicz M., Protokół komunikacyjny Modbus RTU z wykorzystaniem sterownika PLC i układu FPGA – analiza porównawcza, *Przegląd Elektrotechniczny*, R. 96, Nr 2/2021, 93-98, ISSN 0033-2097, doi: 10.15199/48.2021.02.22
- [13] Muc A., Zastosowanie platform cyfrowych Arduino i Raspberry Pi w nauczaniu sterowania obiektem pneumatycznym, *Zeszyty Naukowe WEiA Politechniki Gdańskiej*, Nr 41, 2015, 45-50
- [14] Montironi M. A., Qian B., Cheng H. H., Development and application of the ChArduino toolkit for teaching how to program Arduino boards through the C/C++ interpreter Ch, *Computer Application Engineering Education*, Wiley, 2017, 25, 1053-1065, doi: 10.1002/cae.21854
- [15] Wang Y, Hill K. J, Foley E. C., Computer programming with Python for industrial and systems engineers: Perspectives from an instructor and students, *Computer Application Engineering Education*, Wiley, 2017, 25, 800-811, doi.org/10.1002/cae.21837
- [16] www.randomnerdtutorials.com/25-arduino-shields (dostęp styczeń 2020)
- [17] www.arduino.cc/en/Main/Software (dostęp luty 2019)
- [18] https://www.store.arduino.cc/portenta-h7 (dostęp marzec 2020)
- [19] www.raspberrypi.org/documentation (dostęp marzec 2019)
- [20] www.pcworld.com/article/2085003/intel-ceo-points-toward-wearable-future-with-smart-earbud-smartwatch.html (dostęp grudzień 2020)
- [21] www.software.intel.com/en-us/iot/hardware/edison/documentation, (dostęp grudzień 2020)
- [22] www.terasic.com.tw/en (dostęp grudzień 2019)
- [23] Stalica D., Wysocki W., Projekt i budowa autonomicznego pojazdu rozpoznającego znaki drogowe, praca dyplomowa inżynierska, UMG 2019
- [24] www.gdansk.pl/wiadomosci/bez-kierowcy-i-bez-spalin-dwa-supernowoczesne-busy-juz-w-gdanskua,153475 (dostęp grudzień 2019)
- [25] www.googleblog.blogspot.com/2014/04/the-latest-chapter-for-self-driving-car.html (dostęp grudzień 2020)
- [26] Viola P., Jones M., Rapid Object Detection using a Boosted Cascade of Simple, *IEEE*, 2001
- [27] Derengowski K., Zastosowanie platformy Arduino do sterowania wybranym modelem rzeczywistego obiektu, praca dyplomowa inżynierska, UMG 2021
- [28] https://www.fritzing.org/ (dostęp styczeń 2019)
- [29] https://www.blog.elektroweb.pl/serwo-obsługa-serwomechanizmów-i-zasada-działania (dostęp grudzień 2020)
- [30] https://www.okystar.com/wp-content/uploads/2017/07/OKY3127.pdf, (dostęp grudzień 2020)
- [31] Diya Thomas, Binsu C. Kovoov, A genetic algorithm approach to autonomous smart vehicle parking system, 6th International Conference on Smart Computing and Communications, ICSCC 2017, 2017, India, *Procedia Computer Science*, 125, 2018, 68-76