

Zbuduj własnego robota

Czym jest linefollower? Angielska nazwa składa się z dwóch członów: „line” i „follower”, a więc, jak już się pewnie domyślasz, jest to robot, który śledzi linię. I tak w rzeczy samej jest – linefollowery to klasa robotów, które autonomicznie (to znaczy nie są zdalnie sterowane, nie mają też na stałe zapisanej w pamięci trasy) pokonują tor wyznaczony przez linię. W niniejszym artykule przejdziemy wspólnie przez proces budowy i programowania takiego robota!

Uwaga!

Projekt należy wykonywać w obecności rodzica lub opiekuna!

Komputery na dobre zagościły w naszym życiu. Spotykamy je i wykorzystujemy na każdym kroku, czasami nie będąc nawet tego świadomymi. Dzieje się tak dlatego, że mogą one przyjmować różne formy i mieć różne obudowy. Jeszcze nie tak dawno słowo „komputer” kojarzyło się albo z urządzeniem o wielkości całej sali, albo chociaż dużym białym pudłem na biurku, na którym stał monitor. Początkowo taki duży i ciężki – kineskopowy. Później trochę cieńszy i lżejszy, gdy popularna stała się technologia LCD (ang. *liquid-crystal display*). Przez lata definicja komputera musiała ulec zmianie na skutek postępującej miniaturyzacji i rozwoju techniki. Dzisiejsze telefony komórkowe to dużo bardziej zaawansowane jednostki niż elektroniczny mózg kierujący Apollo 11 podczas pierwszego lądowania człowieka na Księżycu. Komputer jednak wcale nie musi mieć ekranu ani klawiatury. Czasami nie musi stanowić nawet głównego urządzenia, ale może być częścią jakiejś bardziej skomplikowanej maszyny. Wbudowane komputery mogą zarządzać samolotami, kierować samochodami czy sterować pralką. Natomiast maszyny, którymi steruje program komputerowy w celu wykonania jakiegoś zadania, nazywamy robotami. Przykładem takiego robota może być autonomiczny samochód pokonujący trasę z punktu A do punktu B bez ingerencji kierowcy. Oczywiście taki samochód jest urządzeniem bardzo skomplikowanym, łączącym wiele mechanicznych części z ele-

mentami systemu komputerowego. Stworzenie takiego urządzenia wymaga tysięcy godzin pracy różnych zespołów inżynierów oraz dużej ilości pieniędzy zainwestowanych w projekt. Zdecydowanie nie jest to zajęcie dla jednej osoby. Ale nic straconego! Jeżeli chcesz złożyć samodzielnie pojazd, który porusza się bez jakiegokolwiek pomocy ze strony człowieka po z góry wyznaczonej trasie, to zapraszam do lektury tego artykułu.

Bardziej skomplikowane roboty klasy *linefollower enhanced*, oprócz przejazdu trasy, potrafią także pokonywać przeszkody, takie jak przerwy w liniach wyznaczających drogę, rampy czy przedmioty, które należy objechać. Wśród pasjonatów tego typu pojazdów autonomicznych bardzo popularne stały się zawody, których celem jest stworzenie własnego robota od podstaw, który przejedzie trasę w jak najkrótszym czasie. Konkurencja jest bardzo duża, więc roboty wymagają bardzo przemyślanej konstrukcji i ulepszonych programu sterującego, który zagwarantuje szybkie, płynne i bezpieczne pokonanie trasy.

Nasz robot będzie nieco uproszczony, ale zastosowane rozwiązania pozwolą na jego rozbudowę w przyszłości: poprawienie właściwości jezdnych, optymalizację kodu oraz przystosowanie do pokonywania tras z przeszkodami (przeznaczonych dla robotów klasy *enhanced*). Na mojej stronie <https://foxtrotlabs.cc/fl> możesz zgłosić swojego robota, podzielić się jego zdjęciem oraz filmem, który prezentuje przejazd *linefollowera* po trasie.

CZEGO BĘDZIEMY POTRZEBOWAĆ?

Budowanie robotów składa się z dwóch zasadniczych elementów. Pierwszym z nich jest sprzęt (ang. *hardware*)

– budowa fizycznego pojazdu, który będzie przemieszczał się po planszy, a drugim jest oprogramowanie (ang. *software*) – pisanie programu, który będzie nim sterował.

Poniżej wyszczególnione zostały elektroniczne części potrzebne do wykonania robota. Cały koszt budowy nie powinien przekroczyć kwoty 100 PLN.

- » Arduino UNO lub zgodny klon (na przykład Arduino UNO R3 lub Funduino UNO R3) – serce, a właściwie mózg robota – układ wyposażony w mikroprocesor, który będzie odbierał informacje o tym, gdzie jest ścieżka z czujników, i sterował silnikami w taki sposób, aby robot nie zjechał z trasy. Arduino to projekt otwartoźródłowy (open source), dlatego na rynku można znaleźć klony zgodne ze specyfikacją oryginału – są dużo tańsze.
- » listwa z czujnikami linii – oczy robota, które pozwolą ustalić, jak przebiega trasa i jakie jest położenie robota względem niej.
- » mostek-h TB6612FNG – układ, który pozwala sterować silnikami za pomocą mikrokontrolera umieszczonego na Arduino UNO.
- » 2 silniki 65x26mm 5V z przekładnią 48:1 wraz z kołami – dwa niezależne silniki służące zarówno do napędzania robota, jak i sterowania. Nasz robot nie będzie wyposażony w skrętną oś. Zmiana kierunku jazdy będzie się odbywała poprzez sterowanie prędkością obrotową kół. Podobny mechanizm wykorzystywany jest między innymi w czołgach. 😊
- » breadboard – płytki prototypowa, która umożliwia szybkie budowanie układów bez lutowania, dzięki czemu łatwo wprowadzać w nich zmiany.
- » kabel USB-A do USB-B – służy do podłączenia i zaprogramowania Arduino.
- » koszyk na 2 ogniwa 18650 – to one stanowią będą zasilanie robota. Koszyk umożliwia ich łatwe umieszczenie w pojeździe, nie utrudniając przy tym wymiany na naładowane.
- » Stabilizator 5V L7805ABV - THT TO220 – pozwoli zasilić Arduino UNO prosto z koszyka baterii. Alternatywnie możesz wykorzystać powerbank.
- » ogniwa 18650 – zasilanie *linefollowera*. Stosunkowo niedrogi rozwiązanie zapewniające odpowiednie parametry potrzebne do zasilenia robota. Dodatkowo obniżają koszt budowy, ponieważ można je ładować i wykorzystywać wielokrotnie.

- » uniwersalna płytki PCB (ang. *printed circuit board*) – laminat uniwersalny, który nie ma wytrawionych ścieżek, ale umożliwia przylutowanie elementów przewlekanych (technologia THT) oraz ich późniejsze połączenie za pomocą przewodów (tak zwany pająk). Oprócz stworzenia trwałego układu (co odróżnia to rozwiązanie od płytki prototypowej) uniwersalne PCB może stanowić lekką, sztywną i wytrzymałą platformę dla naszego robota.
- » kilka białych kartek formatu A4 – posłużą nam do testowania, czy robot wykrywa linię.
- » czarna taśma izolacyjna – to za jej pomocą wyznaczymy trasę dla *linefollowera*.
- » lutownica – kolbowa lutownica o mocy minimum 45W. Dobrze by było, gdyby pozwalała na regulację temperatury.
- » cyna – ołowiowa cyna grubości 0.5 milimetra, idealna do lutowania układów elektronicznych.
- » trochę goldpinów raster 2,54 – metalowe elementy, które pozwalają przedłużyć wyprowadzenia układów w taki sposób, aby mieściły się na płytce prototypowej, często dodawane do elementów takich jak mostek-h.
- » trzecia ręka – określenie wyposażenia biurka/warsztatu, które pozwala trzymać lutowane elementy. Nie jest to element wymagany, ale znacznie ułatwia pracę.
- » opaski zaciskowe (popularne trytytki) – plastikowe elementy umożliwiające umocowanie części robota.
- » taśma dwustronna – pozwoli unieruchomić silniki.
- » klej na gorąco.
- » podwozie robota – element, do uzyskania którego możesz wykorzystać swoją wyobraźnię. Ja skorzystałem z dwóch patyczków do lodów oraz listewki zakupionej w sklepie ogrodniczym. Podwozie musi być stabilne oraz dać możliwość przymocowania silników, sensorów, koszyka na baterie oraz Arduino UNO.

Program do sterowania *linefollowerem* napiszemy w języku programowania C++ z wykorzystaniem środowiska Arduino. Co to znaczy? Ogólnie rzecz biorąc, Arduino (zobacz link [A] w ramce „W sieci” na końcu artykułu) to platforma służąca do programowania układów z mikrokontrolerami. Dokładnie takimi jak nasza płytki Arduino z mikrokontrolerem ATmega. Wprowadza jednak kilka rozwiązań, które pozwalają szybciej zacząć zabawę

z układami elektronicznymi. Oczywiście niesie to za sobą pewne ograniczenia, ale w naszym projekcie nie będą one przeszkodą. Zainteresowanych zachęcam do zapoznania się ze światem mikrokontrolerów AVR.

ŚRODOWISKO TESTOWE

Niezależnie od tego, czy tworzysz aplikacje webowe, system operacyjny czy program kontrolujący lot rakiety, musi on zostać dobrze sprawdzony w kontrolowanych warunkach, zanim zostanie oddany do użytku. Takie warunki zapewnia specjalnie przygotowane środowisko testowe. W naszym przypadku środowisko testowe będzie stworzone fizyczne, to znaczy, że nie będziemy konfigurować serwerów, ale wykorzystamy przedmioty, które znajdziemy na biurku i w szufladzie – kartkę papieru i taśmę izolacyjną. Po co? W przypadku gdy najpierw skupimy się na odczytach wszystkich fototranzystorów, a potem upewnimy się, że robot może jeździć do przodu, do tyłu, a także skręca zgodnie z tym, jak go

zaprogramujemy, to będziemy mogli napotkać problemy rozwiązywać na bieżąco.

O wiele prościej jest wykrywać błędy w przypadku kontrolowanych warunków i niezbyt skomplikowanych układów. Gdybyśmy złożyli całego robota, zaprogramowali go i okazałoby się, że jeden z czujników został niepoprawnie wpięty, przez co jego odczyty są zaburzone (to znaczy raz bywają prawidłowe, a raz nie) i nasz pojazd raz skręca, a raz porusza się prosto, nie zważając na to, jak biegnie trasa, to wykrycie i rozwiązanie takiego problemu w gotowej konstrukcji jest o wiele trudniejsze. Co jest przyczyną usterki? Czujniki? Napęd?

Przygotowałem kilka fragmentów trasy, które będą pozwalały testować zachowanie robota. Zostały przedstawione na Ilustracji 1.



Ilustracja 1. przykładowe trasy testowe

Twoje kartki testowe nie muszą być identyczne jak te tutaj. Warto jednak pamiętać, aby trasa nie była zbyt skomplikowana. Nasz robot ma pewne rozmiary, porusza się z pewną prędkością, czujniki są od siebie oddalone o określone odległości. To znaczy, że zbyt małe nawroty czy pętle mogą być zwyczajnie za trudne do pokonania. Zauważ też, że przednia część podwozia robota nie ma kółek, a czujniki poruszają się, ocierając o trasę. Zadbaj o to, aby naklejona taśma nie tworzyła żadnych fałd, które mogłyby sprawiać trudności robotowi. W przypadku dużych prędkości osiągniętych na trasie może się to skończyć nawet urwaniem drobnych i delikatnych elementów.

PRZYGOTOWANIE ŚRODOWISKA PROGRAMISTYCZNEGO

Prototypowanie układów z Arduino jest bardzo proste i wymaga jedynie zainstalowania Arduino IDE (Integrated Development Environment) – środowiska programistycznego, które pozwala pisać kod, skompilować go, wgrywać do pamięci mikrokontrolera, a także odbierać informacje przesyłane z płytki Arduino do komputera poprzez port szeregowy.



WARTO WIEDZIEĆ

Czy wiesz, że rozróżniamy dwa główne rodzaje montażu elementów elektronicznych na PCB?

* THT (z angielskiego: *Through-Hole Technology* – technologia montażu przewlekane). Montaż przewlekany polega na tym, że podzespoły elektroniczne mają wyprowadzenia w postaci drucików. Te druciki umieszcza się w otworach na płycie drukowanej i aplikuje się cynę w procesie lutowania po drugiej stronie tej płytki.

* SMT (z angielskiego: *Surface-Mount Technology* – technologia montażu powierzchniowego). To technika, w której podzespoły są montowane po tej samej stronie, na powierzchni laminatu. Nóżki takich układów są mniejsze, a czasami takie układy nie mają wyprowadzeń. Istnieje kilka metod lutowania ich do powierzchni – wykorzystując piny albo kulki BGA (ang. *Ball Grid Array*). Elementy SMD (z angielskiego: *Surface-mount devices*, bo tak nazywamy te podzespoły elektroniczne, które zostały przystosowane do montażu powierzchniowego) ułatwiają automatyzację budowania urządzeń elektronicznych, są mniejsze oraz szybsze w montażu.

Obydwie te technologie mogą być wykorzystywane jednocześnie w urządzeniu i tak zapewne jest w twojej pralce, telewizorze, komputerze czy Arduino UNO wykorzystywanym w naszym projekcie.