

PAWEŁ KARELUS
KRYSTIAN SZOPA

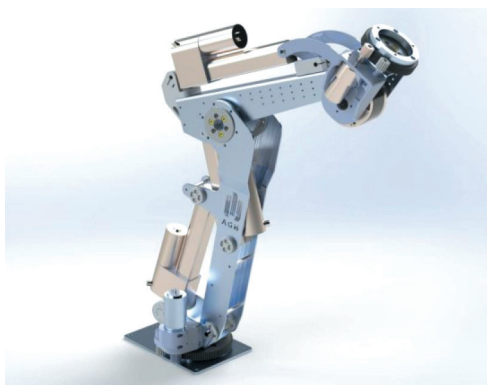
Projekt i wykonanie trójosiowego manipulatora Pick and Place

W artykule przedstawiono projekt i wykonanie trójosiowego manipulatora Pick and Place, który oparty został na oprogramowaniu OpenPnP. Przeznaczony on jest do układania elementów SMD na płytkach PCB. Do wykonania projektu użyto gotowej ramy, która wymagała doboru odpowiednich silników i zaprojektowania wymaganych elementów dla osi Z, które zostały wykonane w programie Solidworks. Do skonstruowania manipulatora zastosowano napęd oparty na przekładni pasowej, jak i elementy pneumatyczne służące do podnoszenia komponentów. Dobrano i skonfigurowano również odpowiednie kamery wraz z oświetleniem, pozwalające na automatyczne sprawdzanie przebiegu pracy maszyny przez oprogramowanie. Kolejnym krokiem był dobór elementów elektronicznych maszyny i ich konfiguracja. Następnie skonfigurowano oprogramowanie OpenPnP i przygotowano je do automatycznego układania elementów SMD.

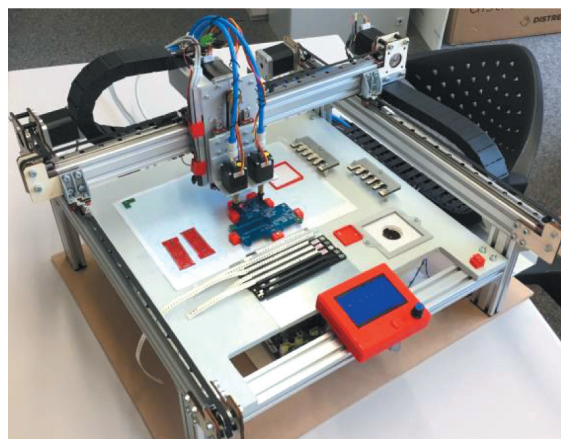
Słowa kluczowe: manipulator, pick and place, SMD, OpenPnP

1. WPROWADZENIE

Manipulatory Pick and Place [1], jak sama nazwa wskazuje, powinny wykonywać czynności związane z podnoszeniem i przenoszeniem komponentów z jednego miejsca na inne. Istnieje wiele rozwiązań takich konstrukcji, a dwa główne rozwiązania to: robotyczna ręka (rys. 1) oraz dwu- lub trójosiowy manipulator montażowy (rys. 2), który został opisany w tym artykule.



Rys. 1. Manipulator w stylu robotycznej ręki zaprojektowany przez AGH Space Systems



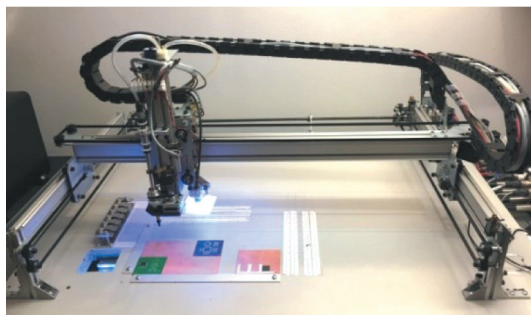
Rys. 2. Prosty trójosiowy manipulator montażowy [2]

Jego zadaniem będzie podnoszenie i układanie elementów SMD (ang. *Surface-mount device*, co tłumaczy się dosłownie jako urządzenie do montażu powierzchniowego. Skrót ten oznacza wszystkie elementy elektroniczne, które umieszczane są na powierzchni płytek PCB, tj. rezystory, kondensatory czy procesory) na płytkach obwodu drukowanego. W zakładach zajmujących się produkcją złożonych

komponentów elektronicznych na skalę masową takie manipulatory wykonują dużą część pracy. Jednak są one przemysłowe, to znaczy, że posiadają stosunkowo duże gabaryty, jak również są bardzo kosztowne w zakupie i utrzymaniu.

Małe zakłady, które wykonują projekty w liczbie niewielu sztuk, mogą zlecać złożenie płytki firmom zewnętrznym, jednak jest to również rozwiązanie drogie, a układanie elementów ręcznie często trwa wiele godzin i jest pracą męczącą z uwagi na jej dużą powtarzalność oraz niewielkie rozmiary elementów. W celu skrócenia czasu potrzebnego na ułożenie elementów można zastosować manipulator Pick and Place, który można zakupić lub też wykonać samemu. Pozwala to na uwolnienie pracownika. Może on w tym czasie wykonywać bardziej złożone zadania, kiedy maszyna wykonywać będzie prostą, powtarzalną i długo trwającą pracę.

Istnieje wielu komercyjnych producentów manipulatorów Pick and Place. Dostępne są konstrukcje o różnych wielkościach i różnej przepustowości, od mniejszych, mających pojedyncze głowice, do większych przemysłowych konstrukcji, mających kilkanaście, a nawet kilkadziesiąt głowic [3–4]. Jednym z przykładów takiej maszyny dostępnej na rynku jest manipulator LitePlacer (rys. 3) produkowany przez fińską firmę o tej samej nazwie.



Rys. 3. Manipulator LitePlacer [3]

Dostępne są również konstrukcje przemysłowe mające bardzo dużą prędkość pracy i mogące układać dziesiątki tysięcy elementów na godzinę oraz mające setki wejść na różne elementy SMD, co jest potrzebne, jeżeli projekty płytek są skomplikowane. Służą one jednak do produkcji seryjnej i mają zastosowanie tylko w dużych zakładach. Jednym z dostępnych rozwiązań jest maszyna JUKI RS-1R (rys. 4), produkowana przez chińską firmę ETA. Taki manipulator stosowany jest jako część linii produkcyjnej, posiada automatyczne wejścia i wyjścia w postaci taśmociągów, dzięki którym można połączyć kilka maszyn wykonujących kolejne punkty procesu produkcyjnego.

Jego przepustowość wynosi około 47 000 części na godzinę, co wydaje się ogromną liczbą.



Rys. 4. Manipulator JUKI-RS-1 [4]

Jeżeli jednak koszty związane z kupnem manipulatora są zbyt wielkie, istnieje możliwość zaprojektowania własnego urządzenia z wykorzystaniem dostępnych darmowych środowisk do sterowania. Jest to jednak rozwiązanie czasochłonne i wymagające znajomości mechaniki i budowy maszyn, elektroniki oraz podstaw programowania.

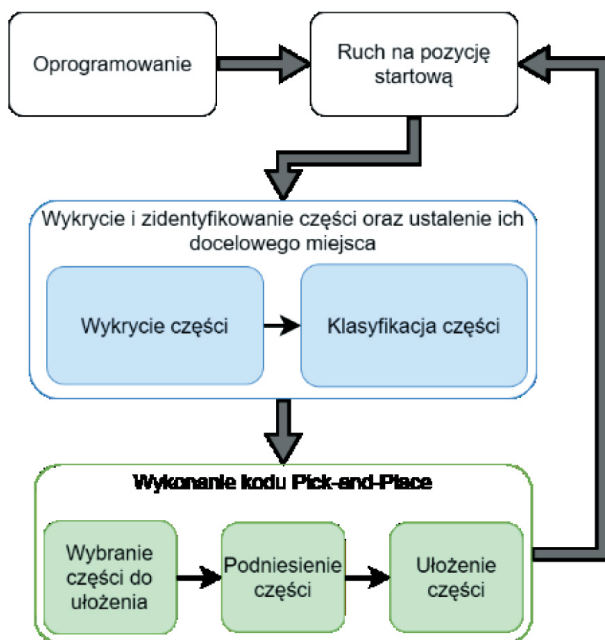
2. HISTORIA I CHARAKTERYSTYKA MANIPULATORÓW PNP

Historia manipulatorów Pick and Place sięga lat 30. XX wieku, jednak dopiero po drugiej wojnie światowej zostały stworzone pierwsze przemysłowe jednostki. Wraz z rozwojem technologii i ekonomii zapotrzebowanie na liczbę produkowanych przez zakłady artykułów znacząco rosło, zwiększyło to zainteresowanie maszynami, które będą wykonywały proste i powtarzalne czynności zamiast człowieka, zwłaszcza w środowisku zagrażającym zdrowiu lub życiu pracowników [5].

Algorytm pracy manipulatorów montażowych na pierwszy rzut oka nie jest skomplikowany. Wystarczy wykonać tylko kilka podstawowych kroków, które należy wielokrotnie powtórzyć. W praktyce skalibrowanie każdego kroku wymaga czasu z powodu wymaganej precyzji oraz powtarzalności.

Algorytm sterowania manipulatorem Pick and Place wymaga kilku kroków (rys. 5). Zacząć należy od projektu CAD płytki PCB z rozmieszczonymi w nim elementami, które mają zostać rozłożone. Następnie projekt ten wczytywany jest przez oprogramowanie zawarte na komputerze służącym jako główny sterownik urządzenia. Program ten dopasowuje poszczególne elementy na płytce do elementów na podajnikach oraz tworzy instrukcje w języku zrozumiałym dla sterownika elektronicznego, najczęściej jest to G-code.

Sterownik otrzymuje instrukcje i wykonuje je przez wysyłanie impulsów elektrycznych na odpowiednie złącza, co powoduje ruch maszyny w odpowiednich kierunkach, umożliwiając realizację zamierzonego celu. Do wykrywania poprawnego ułożenia podniesionych komponentów najczęściej używane są kamery, z których obraz jest przetwarzany przez oprogramowanie w odpowiedni sposób, który zostanie opisany w dalszej części artykułu.



Rys. 5. Schemat pracy manipulatora PnP [6]

3. OPIS WYKONANEJ KONSTRUKCJI

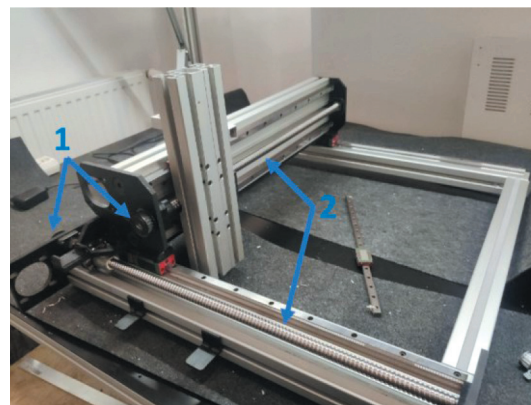
3.1. Przygotowanie projektu

Początek każdego projektu stanowią założenia oraz cele. W tym przypadku zostanie zastosowana wcześniej przygotowana rama (rys. 6). Pozwoliło to na zmniejszenie kosztów potrzebnych na skonstruowanie manipulatora, jak również skrócenie czasu na to poświęconego. Konstrukcja ta posiadała już gotowe osie X i Y z zastosowanym napędem śrubowym. Przyjęte etapy pracy przedstawiały się następująco:

- wybranie i zamontowanie silników napędzających osie,
- dobór przełożenia przekładni pasowych służących do napędzania osi X i Y,
- projekt oraz wykonanie osi Z wraz z mocowaniami na dwie głowice,
- wykonanie uchwytów na czujniki,
- montaż kamer,
- wykonanie okablowania maszyny,

- montaż manipulatora do stolika z kółkami,
- montaż tacek z elementami SMD,
- montaż zasilacza, kontrolerów, sterownika manipulatora oraz awaryjnego wyłącznika,
- poprawne podłączenie oraz konfiguracja sterownika za pomocą oprogramowania przeznaczonego dla drukarek 3D,
- konfiguracja oprogramowania OpenPnP.

Na początku więc zostanie wykonana konstrukcja mechaniczna manipulatora, następnie zostaną podłączone elementy elektroniczne, a ostatnim etapem będzie zainstalowanie i konfiguracja oprogramowania. Zastosowanie dwóch głowic motywowane jest tym, że przy planowanym użyciu napędu opartego o przekładnię pasową sterowanie nimi przebiega w prosty sposób. Zamocowanie ich po dwóch stronach jednego koła pasowego powoduje, że kiedy jedna głowica poruszana jest w jedną stronę, to druga porusza się w stronę przeciwną. Zastosowanie dwóch głowic oznacza też szybszą pracę manipulatora, niż kiedy zastosowana byłaby tylko jedna. Rysunek 6 pozwala zapoznać się z ogólną budową ramy.



Rys. 6. Początkowy wygląd zastosowanej ramy:

1 – napędy śrubowe; 2 – koła pasowe

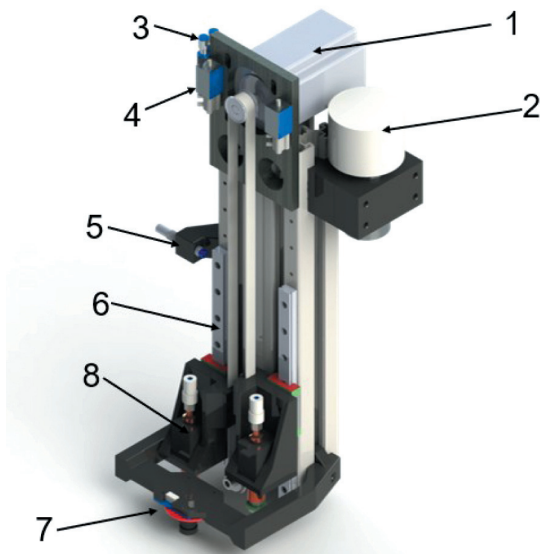
3.2. Konstrukcja mechaniczna

Z racji tego, że do skonstruowania manipulatora użyto gotowej ramy, niektóre elementy były wybrane wcześniej. Do takich elementów należy m.in. napęd śrubowy w kierunku osi X oraz Y, czy przenoszenie momentu z silnika, które odbywa się poprzez przekładnię pasową (oba elementy widoczne są na rys. 6). Jako że napędy te nie będą przenosić dużych momentów, a wymagana jest jak największa prędkość, dobrano przełożenie przekładni równe 1. Silniki dobrano tak, aby rozstaw montażowy otworów pokrywał się z otworami wykonanymi w ramie. Wybrano silniki

krokowe z serii Nema 24 – 60HS88 o momencie nominalnym 3 Nm. Koła pasowe zostały dobrane, biorąc pod uwagę te już zamontowane na śrubach napędowych, aby otrzymać założone przełożenie. Jedyнным możliwym sposobem okazał się dobór identycznej wersji kół, a mianowicie kół pasowych HTD 22-5M-09 posiadających 22 zęby i średnicę podziałową równą 35 mm. Osie zamocowane są do prowadnic liniowych na przeznaczonych specjalnie do nich wózkach, aby osiągnąć jak najmniejszy opór ruchu i stabilność konstrukcji.

3.3. Projekt osi Z

Projekt osi Z (rys. 7) został wykonany w programie Solidworks według przedstawionych wcześniej założeń. Wymagało to zamontowania silnika na osi wraz z przekładnią pasową, która ma napędzać głowice, które zostały zamocowane w specjalnie wydrukowanych uchwytach na prowadnicach liniowych oraz połączone bezpośrednio z przekładnią pasową. Spowodowało to naprzemienny ruch głowic. Na osi znajduje się też system pneumatyczny wywołujący podciśnienie w głowicy, umożliwiające podnoszenie elementów SMD.



Rys. 7. Projekt układu napędu w kierunku osi Z wykonany w programie Solidworks:

1 – silnik krokowy; 2 – pompka; 3 – rozdzielacz pneumatyczny; 4 – elektrozawór; 5 – indukcyjny czujnik krańcowy; 6 – prowadnica liniowa; 7 – kamera górna; 8 – głowica

Układ ten składa się z następujących elementów: pompki próżniowej, rozdzielacza, dwóch elektrozaworów pneumatycznych o jednym wejściu i dwóch wyjściach oraz przewodów pneumatycznych. Zada-

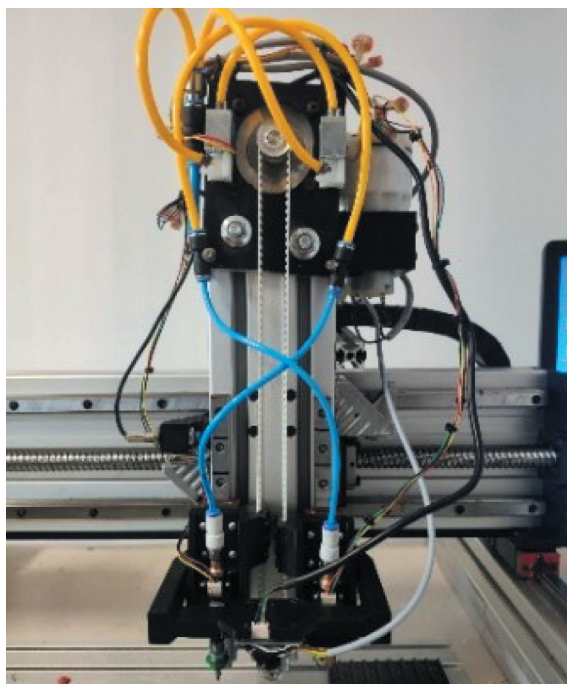
niem pompki jest wytwarzanie podciśnienia w przewodach, umożliwiającego podnoszenie elementów SMD. Głowica wykonana jest z wału drążonego pozwalającego jednocześnie na obrót końcówki wraz z elementem podniesionym za pomocą wspomnianego wcześniej podciśnienia. Do poprawnego funkcjonowania oprogramowania potrzebna jest też kamera, która będzie w stałej pozycji XY względem głowic. Umożliwia ona zarówno wykrywanie elementów na taśmach, jak i pozycji płytek PCB.

Na górnej części profilu aluminiowego został zamontowany zarówno silnik, jak i system pneumatyki. Silnik został zamocowany w taki sposób, aby możliwe było zamocowanie na jego trzpieniu koła zębatego, a samo mocowanie silnika umożliwia naciągnięcie paska. Użyto elektrozaworów z uwagi na to, że została zamocowana jedna pompka, jeżeli elementy mają być układane jeden po drugim, wyłączenie pompki skutkowałooby utratą wytworzonego podciśnienia w obu głowicach, co powodowałoby upuszczenie obu elementów. Przełączenie zaworu zamiast wyłączenia pompki pozwala na wyrównanie ciśnienia tylko w jednej głowicy. Rozdzielacz umożliwia podłączenie dwóch zaworów do jednej pompki. W modelu CAD nie zostały poprowadzone przewody pneumatyczne z uwagi na to, że są one podłączane luźno, więc trudno przewidzieć ich ułożenie.

Indukcyjny czujnik krańcowy (pozycja 5 na rys. 7) mocowany jest tak, aby można było w każdym momencie dostosować jego pozycję. Mocowanie do profilu aluminiowego odbywa się za pomocą śruby z nakrętką zamkową, co wraz ze specjalnie zamontowanym uchwytem powoduje sztywną pozycję po dokręceniu mocującej śruby. Sam czujnik posiada gwint na prawie całej swojej długości, co również daje możliwość dostosowania odpowiedniej pozycji.

Głowice (pozycja 8, rys. 7) odpowiedzialne za pracę podnoszenia i odkładania elementów montowane są w specjalnie zaprojektowanych uchwytach i mocowane za pomocą śrub do wózków na prowadnicach liniowych. Poprzez taką konstrukcję ich ruch odbywa się płynnie i bez problemów związanych z tarcieniem, dzięki połączeniu uchwytu do paska zębatego za pomocą specjalnie zaprojektowanego mocowania. Takie połączenie elementów ogranicza ruch przekładni do niewielkiego zakresu, jednak wymagany ruch głowicy jest niewielki.

Górna kamera (pozycja 7, rys. 7) powinna znajdować się w stałej odległości od głowic, dlatego również została zamocowana za pomocą odpowiednio zaprojektowanego uchwytu.



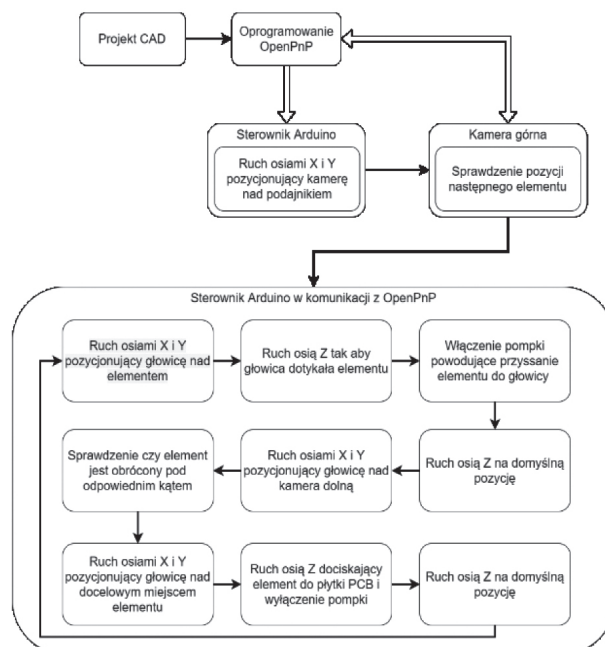
Rys. 8. Końcowy wygląd układu napędu w kierunku osi Z wraz z układem podnoszącym

Końcowy wygląd konstrukcji osi Z przedstawiony został na rysunku 8. Same przewody podłączone są krzyżowo, tj. lewy zawór podłączony jest do prawej złączki, a prawa złączka podłączona została do lewej głowicy. Rozwiązanie to powoduje większą swobodę ruchu przewodów podczas pracy głowicy oraz minimalizuje ostre zagięcia przewodów. Elementy SMD w większości nie posiadają dużej masy, jednak aby mieć pewność, że każdy większy mikroukład, jak np. procesor, został podniesiony prawidłowo, należało zadbać o jak największą wartość podciśnienia na końcu głowicy.

3.4. Algorytm sterowania

Sterowanie ruchami maszyny odbywa się automatycznie lub manualnie. Sposób manualny jest przewidziany do konfiguracji, każdy krok można wykonać samemu, uruchamiając poszczególne opcje w oprogramowaniu OpenPnP, jednak domyślnym sposobem jest sterowanie automatyczne. Aby było ono możliwe, wymagane jest wcześniejsze wprowadzenie wszystkich potrzebnych danych do programu i sterownika. Algorytm sterowania (rys. 9) przedstawiony jest tylko w przypadku sterowania automatycznego. Ruchy maszyny odbywają się na podstawie instrukcji wysyłanych w języku G-code. Instrukcje tworzone są przez program

OpenPnP wysyłane do mikrokontrolera Arduino, który konwertuje kod i przekazuje sygnały do odpowiednich komponentów maszyny, powodując ich ruch. Kod wysyłany jest na bieżąco, wymagane więc jest połączenie sterownika z komputerem przez cały okres pracy maszyny.



Rys. 9. Algorytm sterowania manipulatora

Nie wszystkie elementy są jednak podłączone bezpośrednio do sterownika. Kamera górna i dolna podłączana jest przez port USB do komputera obsługującego maszynę, aby oprogramowanie OpenPnP miało dostęp do obrazu i mogło odpowiednio dobrać parametry odległości czy obrotu, o który trzeba przesunąć głowicę. Drugim takim elementem są światła LED zamontowane przy kamerach. Służą one do oświetlenia płytek i taśm z elementami SMD w przypadku kamery górnej oraz elementów przyssanych do głowicy w przypadku kamery dolnej. Podłączone są one do sterownika Arduino Nano, który następnie podpięty jest pod odpowiednie piny na głównym sterowniku. Sterowanie światłami opisane zostanie w dalszej części pracy.

4. ELEMENTY ELEKTRONICZNE I ICH OPROGRAMOWANIE

Do konstrukcji prostych manipulatorów PnP najczęściej wykorzystuje się elementy stworzone do obsługi drukarek 3D, są one łatwo dostępne, nie są kosztowne

oraz stosunkowo łatwo można je skonfigurować do obsługi manipulatora. Jednym z najważniejszych praktycznych wymagań jest to, aby oprogramowanie sterownika było w stanie obsługiwać oryginalnie dwa ekstrudery w drukarce, jeżeli jednak tak nie jest, można zmienić kod konfiguracyjny w taki sposób, aby manipulator był obsługiwany, jednak jest to zdecydowanie utrudnione. Biorąc to pod uwagę, skorzystano z mikrokontrolera MEGA 2560 wraz z nakładką RAMPS 1.4. Nakładka ta jest rozszerzeniem mikrokontrolera MEGA 2560, umożliwia ona podłączenie silników krokowych wraz ze sterownikami silników krokowych. Wybrano prosty i niedrogi sterownik A4988, którego potrzebne są docelowo dwie sztuki, jeden sterownik obsługuje tylko jeden silnik, jednak nie są one w stanie sterować silnikami osi X, Y oraz Z. Ich zadaniem jest obsługiwanie osi obrotu. Jeden z tych sterowników jest zamontowany z myślą o przyszłej rozbudowie maszyny. Do sterowania silnikami głównych trzech osi wybrano dwufazowy sterownik HY-DIV268N-5A. Ma on możliwość prostej zmiany natężenia prądu oraz liczby mikrokroków. Taka konfiguracja pozwala na bezproblemowe sterowanie dobranymi silnikami krokowymi. RAMPS daje również możliwość podłączenia czujników krańcowych, z których również skorzystano w tym projekcie. Korzystając z pozostałych pinów, można sterować dodatkowymi elementami, w przypadku manipulatora są to elektrozawory, pompka czy światła LED, do których potrzebne jest jeszcze Arduino Nano, ponieważ sygnał wychodzący z mikrokontrolera jest sygnałem PWM (*Pulse-Width Modulation* [7]). Mikrokontroler wysyła wartości zaczynające się od 0 aż do 255, jest to więc sygnał cyfrowy, który nie jest w stanie zasilić światła. Sygnał wysyłany jest więc do mikrokontrolera Arduino Nano, który po otrzymaniu wartości 0 zapala światła, a przy otrzymaniu wartości większej gasi je. W praktyce te trzy elementy, nie wliczając Arduino Nano, tworzą całość. Nie jest wymagana konfiguracja każdego elementu osobno, oprogramowanie sterownika drukarki 3D wgrywane jest na główny mikrokontroler, tj. Mega 2560, który steruje resztą bez potrzeby zewnętrznej konfiguracji. Dodając do tej konfiguracji dodatkowy mikrokontroler, otrzymujemy możliwość rozszerzenia manipulatora o dodatkowe elementy, w przypadku tej maszyny są to światła LED, jednak w przyszłości można rozbudować konstrukcję, podłączając do niego również np. automatyczne podajniki elementów SMD.

4.1. Konfiguracja mikrokontrolera MEGA

Najważniejszymi zmianami w domyślnej konfiguracji mikrokontrolera było:

- wybranie szybkości transmisji komunikacji sterownika z komputerem,
- ustawienie minimalnej i maksymalnej wartości pozycji poszczególnych osi,
- poprawne ustawienie czujników krańcowych (normalnie otwarte, oś X i Y – czujnik minimalny, oś Z – czujnik maksymalny),
- ustawienie rozdzielczości silników krokowych oraz ich maksymalnej prędkości i przyspieszenia.

W związku z tym, że zastosowane oprogramowanie przeznaczone jest dla drukarek 3D, przewidziane w nim zostały tylko trzy osie. Aby można było użyć czwartej osi, która działa jako obrót głowicy, została ona podłączona do złącza ekstrudera, co wymagało wyłączenia obsługi czujników temperatury stosowanych w drukarkach.

Po ustawieniu wszystkich opcji gotowy program został zainstalowany na sterowniku. Następnym krokiem było skonfigurowanie mikrokontrolera Arduino Nano.

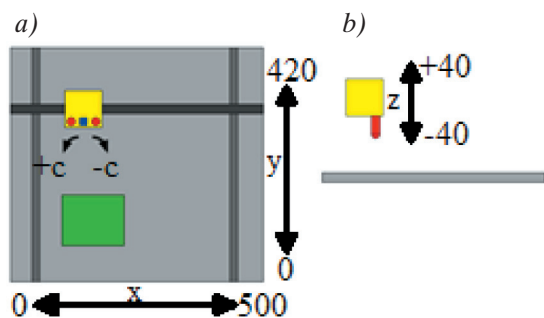
4.2. Konfiguracja mikrokontrolera Arduino Nano

Płytki z diodami LED są przymocowane do uchwytów kamer. Ich zadaniem jest oświetlanie manipulatora, kiedy oprogramowanie pobiera obraz. Z racji tego, że zamontowane są one równolegle do siebie, lecz w przeciwnych kierunkach, nie powinny być one zapalone przez cały czas pracy maszyny. Światło oświetlające obraz dolnej kamery może skutkować zakłóceniami występującymi na obrazie pochodzącym z górnej kamery i odwrotnie. Z tego powodu konfiguracja mikrokontrolera nie jest wystarczająca, aby uzyskać poprawną pracę światła, jednak jest to pierwszy krok. Do pinów 7 oraz 8 urządzenia podłączono kable przenoszące sygnał włączający. Dzięki zastosowaniu odpowiedniego kodu, można sterować zapaleniem i gaszeniem światła, podając niskie i wysokie napięcie. Płytki z diodami LED połączone są szeregowo. Na każdej znajduje się 16 diod. Zastosowanie prostych instrukcji warunkowych pozwala na włączenie całej płytki praktycznie jednocześnie, jeżeli napięcie na wyjściu zmieni się na niskie. Kolor podawany jest w formacie RGB – wartość 250, 0, 0 oznacza, że światła świecą się na kolor czerwony. Odcień

ten został wybrany dlatego, iż żadna część konstrukcyjna manipulatora nie jest w tym samym kolorze oraz powoduje zauważalnie mniejsze odbicia światła niż przy zastosowaniu światła białego.

5. KONFIGURACJA OPROGRAMOWANIA OPENPNP

Zapewnienie poprawnego działania manipulatora wymaga również odpowiednio skonfigurowanego oprogramowania nim sterującego. Na początku sprawdzono, czy wszystkie osie i czujniki działają poprawnie. Środowisko OpenPnP wymaga układu współrzędnych maszyny przedstawionego na rysunku 10. Gdy upewniono się, że elementy konstrukcji działają poprawnie, przystąpiono do wyznaczenia kroków na milimetr dla silników krokowych.



Rys. 10. Układ współrzędnych manipulatora:
a) widok z góry; b) widok z boku [8]

Rozdzielczość silników krokowych można bardzo łatwo obliczyć dzięki podzieleniu liczby 360 przez kąt kroku silnika [9], który wynosi $1,8^\circ$. Obliczona w ten sposób wielkość określa liczbę potrzebnych kroków do pełnego obrotu silnika. Aby obliczyć jednak wartość kroków na milimetr, należy skorzystać ze wzoru dla napędów śrubowych:

$$k_r = \frac{l_k \cdot m_k}{P} \quad (1)$$

gdzie:

- l_k – liczba kroków silnika na pełen obrót,
- m_k – liczba mikrokroków,
- P – skok gwintu śruby napędowej.

Przyjęto 16 mikrokroków. Do wzoru potrzebna jest znajomość również skoku gwintu śruby napędowej, który wynosi 8 mm. Znając wszystkie te wartości, możemy je podstawić do wzoru (1), otrzymując:

$$k_r = \frac{200 \cdot 16}{8} = 400 \quad (2)$$

Wynika z tego to, że przemieszczenie się osi o 1 mm wymaga wykonania przez silnik 400 kroków. Aby mieć pewność, czy wartość została poprawnie obliczona, należy ją jednak sprawdzić w praktyce.

Do kalibracji podstawowych ustawień oprogramowania OpenPnP skorzystano z instrukcji dostępnej w serwisie Github [10] opracowanej przez twórców.

5.1. Początkowe ustawienia oprogramowania

Pierwszym krokiem jest instalacja oprogramowania OpenPnP. Rozpocząć należy od wybrania sterownika oprogramowania, aby mogło ono poprawnie komunikować się z mikrokontrolerem. Dla wszystkich konstrukcji, w których sterowanie opiera się na języku G-code, przeznaczony jest sterownik „Gcode-Driver”. Następnie należy wybrać port oraz prędkość równą 115 200 Bd (bod, ang. *Baud* – miara określająca prędkość przesyłania zmian medium transmisyjnego).

5.2. Kalibracja górnej kamery

Pierwszym kalibrowanym elementem jest górna kamera, która zamontowana jest na osi Z. W kamerach podłączanych przez porty USB zalecane jest użycie sterownika OpenPnP Capture Camera i taki właśnie został wybrany. Jest to kod umożliwiający poprawny odczyt obrazu kamery przez oprogramowanie pozwalający również na jej kontrolowanie. Ustawiono również format obrazu. Parametry przetwarzania obrazu kamery, np. jasność, nasycenie czy balans bieli zostały dobrane tak, aby o każdej porze dnia obraz wyglądał podobnie. Pozwala to na poprawne działania algorytmów przetwarzających obraz, np. w celu wykrycia elementu SMD na taśmie bez względu na to, czy manipulator znajduje się w oświetlonym pomieszczeniu czy nie.

Górną kamerę, dla najlepszego zobrazowania, nazwano *Camera Top*. Jednym z najważniejszych ustawień jest kalibracja jednostek (w tym przypadku milimetrów) na piksel obrazu. Jeżeli parametr ten jest niepoprawnie zmierzony, może to skutkować błędną pracą maszyny, ponieważ w niektórych krokach algorytmu pracy mierzona jest odległość rzeczywista na podstawie obrazu. Obliczone wartości wynoszą 0,0662 mm na piksel obrazu. Górna kamera jest dla oprogramowania punktem zero. Kolejnymi ustawieniami są transformacje obrazu pochodzącego z kamer.

Zgodnie z zaleceniami obraz z górnej kamery powinien wyglądać jak to, co widzi osoba patrząca z góry na manipulator, czyli w górnej części obrazu powinny znajdować się elementy bardziej wysunięte w osi X. Jeżeli zamocowana kamera nie dostarcza takiego obrazu, można odwrócić go w pionie, w poziomie lub obrócić o zadany kąt. Można również wykadrować obraz, jest to operacja przydatna, ponieważ obiektyw kamery przypomina półsferę. Powoduje to, że na obrazie nie jest zachowana stałość wymiarów (liczba pikseli na milimetr w okolicy środka obrazu nie jest równa liczbie pikseli na milimetr przy krańcach obrazu).

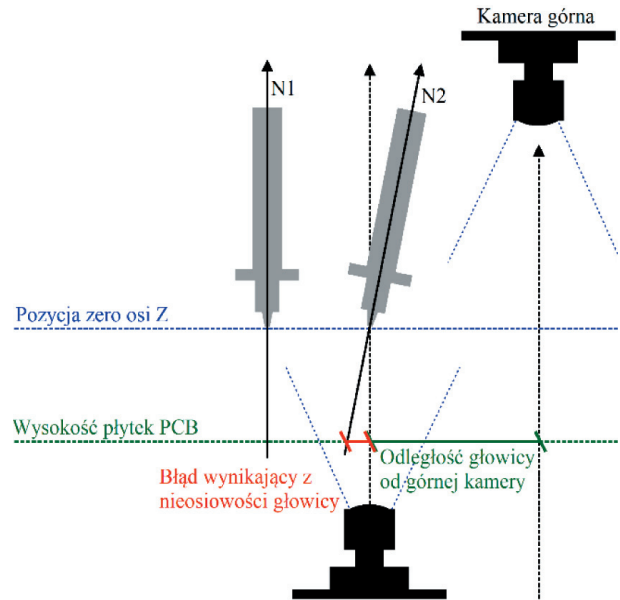
5.3. Kalibracja dolnej kamery

Konfiguracja dolnej kamery przebiega podobnie do konfiguracji kamery górnej, jednak ustawione wartości różnią się, ponieważ element ten ustawiony jest w innym miejscu i odpowiedzialny jest za wykrywanie obrotu podniesionych już części oraz pozycji końcówki głowicy. Kamery nazwano *Bottom Camera* (kamera dolna), aby zachować motyw nazewnictwa. Obliczono również wartość milimetr na piksel obrazu równą 0,017 mm. Ustawienia transformacji obrazu zostały również dobrane w ten sposób, aby był on podobny w każdych warunkach zewnętrznego oświetlenia. Jednym z ważniejszych ustawień dolnej kamery jest ustalenie takiej jej pozycji, w której końcówka głowicy znajduje się idealnie na środku wizji. Nie da się tego obliczyć bez wcześniejszego obliczenia odległości, w jakiej znajduje się głowica od kamery górnej będącej punktem zero dla oprogramowania.

5.4. Kalibracja głowicy

Na początku skonfigurowano tylko jedną głowicę, aby łatwiej można było sprawdzić poprawność działania konstrukcji. Nadano jej nazwę N1. Następnie, aby umożliwić automatyczną obsługę elementów pneumatycznych, dodano odpowiednie włączniki, które wysyłają komendy w języku G-code, powodujące włączenie lub wyłączenie odpowiednich podzespołów. Utworzono więc włącznik *Vaccum* (próżnia), który powoduje włączenie lub wyłączenie pompki oraz dwa włączniki N1_VAC oraz N2_VAC (oznaczające elektrozawory dla głowicy pierwszej i drugiej). Następnym krokiem konfiguracji było określenie, w jakiej

odległości głowica znajduje się od punktu zero manipulatora. Wykonanie tej czynności pozwoliło również na określenie miejsca znajdowania się dolnej kamery. Określenie pozycji kamery pozwoliło również na ustalenie błędu nieosiowości maszyny, który jeżeli jest zbyt duży, może wpłynąć na nieprawidłowe wykonywanie pracy przez manipulator.



Rys. 11. Wizualizacja błędu nieosiowości [11]

Jak widać na rysunku 11, jeżeli głowica nie jest zamocowana w osi prostopadłej do płaszczyzny płytek PCB, skutkuje to przesunięciem. Dodatkowo, jeżeli sama głowica nie jest wykonana z zachowaniem współosiowości albo końcówka głowicy zamontowana jest w sposób powodujący nieosiowość, trudne jest określenie punktu, w którym końcówka głowicy dotknie płytki PCB, ze względu na to, iż przesunięcie zmienia kierunek i/lub wartość razem z obrotem silnika głowicy. Do tego procesu używany jest więc algorytm przetwarzający obraz z kamery i wykrywający końcówkę głowicy. Po ustaleniu pozycji dolnej kamery końcówka głowicy powinna znajdować się w centrum, następnie algorytm wykonuje obroty silnikiem głowicy tak, aby ich suma wyniosła 360°. Daje to pogląd, jak zmienia pozycję końcówka głowicy pod względem obrotu w każdym miejscu. Po wykonaniu każdego z pojedynczych ruchów obrotowych pobierany jest widok z kamery i przetwarzany przez specjalnie dobrane algorytmy tak, aby wykrywał, w jakim miejscu znajduje się powierzchnia styku. Składa się on z kilku mniejszych etapów, które finalnie powinny doprowadzić do wykrycia jednego okręgu pokrywają-

cego się z końcówką głowicy. Dzięki temu wraz z obliczoną wcześniej liczbą milimetrów na piksel kamery dolnej algorytm jest w stanie obliczyć, ile wynosi błąd nieosiowości przy danym obrocie silnika. Wynik ten jest zapisywany, a następnie dodawany lub odejmowany przy wykonywaniu automatycznych ruchów. Należy jednak wziąć również pod uwagę fakt, iż w osi obrotu głowicy nie jest zamocowany żaden czujnik krańcowy. Powoduje to ustawianie pozycji obrotu do wartości 0 przy każdym włączeniu maszyny, przez co dodawane wartości mogą nie działać poprawnie i zwiększać błąd, zamiast go niwelować. Ważne więc jest, aby kalibracja ta odbywała się po każdym bazowaniu manipulatora.

5.5. Bazowanie manipulatora

Po włączeniu zasilania maszyny pozycja osi ustawiana jest automatycznie do wartości 0. Aby oprogramowanie znalazło więc dokładną pozycję maszyny, należy wykonać operację bazowania. Składa się ona z trzech etapów:

1. dojazd osiami do krańcówek według ręcznie wpisanego kodu G-code,
2. dojazd kamerą nad specjalny punkt znajdujący się na stole o niezmienniej pozycji, aby zapewnić jak największą powtarzalność bazowania,
3. dojazd głowicą nad dolną kamerę, aby dokonać kalibracji błędu nieosiowości (opisane w poprzednim podrozdziale).

Etap pierwszy składa się z kilku komend w języku G-code. Zapewniają one bezpieczny dojazd każdej z osi do czujników krańcowych. Przy automatycznym bazowaniu należy pamiętać, że może zdarzyć się sytuacja, kiedy głowica będzie znajdowała się w pozycji niskiej.

Elementy SMD są często niewielkich wymiarów, należy więc korzystać z każdej dostępnej opcji zwiększenia precyzji. Do jednej z tych opcji zaliczyć można ruch kamery nad specjalnie przygotowany czarny punkt znajdujący się na stole o niezmienniej pozycji w stosunku do konstrukcji. W pliku konfiguracyjnym została zawarta jego pozycja, więc po wykonaniu pierwszego etapu bazowania i przesunięciu nad ten punkt jeden z algorytmów przetwarzania obrazu kamery sprawdza, czy znajduje się on idealnie pod kamerą. Jeżeli nastąpi przesunięcie, pozycja maszyny jest korygowana. Kiedy punkt nie zostanie wykryty,

operacja bazowania zostaje przerwana i należy wykonać ją jeszcze raz. Następnym etapem jest dojazd głowicą nad dolną kamerę w celu obliczenia błędu nieosiowości. Jeśli jednak nie zostanie ona poprawnie obliczona, operacja bazowania również zostaje przerwana.

6. ALGORYTMY PRZETWARZANIA OBRAZU

Aby możliwa była automatyczna praca manipulatora, wymagane są odpowiednie algorytmy, które wykrywać będą potrzebne elementy na obrazie pochodzącym z kamer. Jeden algorytm składa się z kilku mniejszych algorytmów, które można dobierać zgodnie z wymaganiami, finalnie doprowadzając do wykrycia elementu. Oczywiście różnią się one w zależności od tego, jaki komponent trzeba wykryć. Dobranie odpowiednich algorytmów powinno być przeprowadzone po konfiguracji obrazu pochodzącego z kamer. Ważne jest, aby algorytmy pracowały podobnie w różnych warunkach oświetlenia.

Ogólna zasada działania algorytmów obrazu jest jednak podobna. Każdy z nich zaczyna od pobrania klatki pochodzącej z obrazu kamery po upływie czasu ustalania. Następnie zmniejsza pobraną klatkę w celu zminimalizowania ilości elementów znajdujących się na obrazie. Kolejno wykonywane są operacje przetwarzania obrazu i końcowym efektem jest narysowanie na pobranej początkowo klatce wykrytego elementu – prostokąta lub okręgu. Jednak algorytmów stosowanych przez oprogramowanie do obsługi kamery jest wiele z powodu tego, że służy ona do wykrywania podniesionych elementów SMD. Algorytmy wykorzystywane przez kamerę dolną można podzielić na dwa główne kierunki, ponieważ te stosowane do wykrywania podniesionych części wyglądają bardzo podobnie, w większości różnią się tylko wyborem pojedynczych wartości w etapach procesu wykrywania.

Drugi algorytm służy do wykrywania końcówki głowicy. Składa się on z ośmiu etapów przedstawionych i opisanych poniżej. Powinny one być dobrane w taki sposób, aby wykrywany był tylko jeden okrąg znajdujący się dokładnie na końcu głowicy. Dopuszczalne są jednak niewielkie odchylenia, jeżeli na przykład wykryty okrąg jest mniejszy lub większy od końcówki głowicy, ale jego środek pokrywa się z rzeczywistym

środkiem elementu, nie powoduje to problemów, ponieważ oprogramowanie bierze pod uwagę tylko punkt środkowy. Końcówka głowicy jest jedna, więc na obrazie nie powinno znajdować się więcej narysowanych okręgów. Etapy algorytmu przedstawiają się następująco [9]:

1. ImageCapture – po upływie czasu ustalania kamery pobierana jest jedna klatka pochodząca z obrazu dolnej kamery.
2. MaskCircle – algorytm ten ogranicza pobraną wcześniej klatkę kołem o średnicy 350 pikseli, pozostała część obrazu pokrywana jest czarnym kolorem.
3. Bgr2Gray – wykonywana jest konwersja kolorów z systemu BGR (24-bitowy zapis wartości koloru) na system odcieni szarości.
4. Threshold – algorytm ten zamienia obraz na dwukolorowy – czarny i biały.
5. BlurMedian – wykonywane jest uśrednianie pikseli. Powoduje to usunięcie szumów z obrazu.
6. DetectCirclesHough – algorytm ten stosowany jest do wykrywania okręgów o średnicy mieszczącej się w zadanym zakresie znajdujących się na obrazie.
7. ImageRecall – algorytm ten nadpisuje przetworzony obraz na początkowo pobraną klatkę.
8. DrawCircles – algorytm ten rysuje wykryte w etapie 6 okręgi na przywołanej w poprzednim kroku klatce.

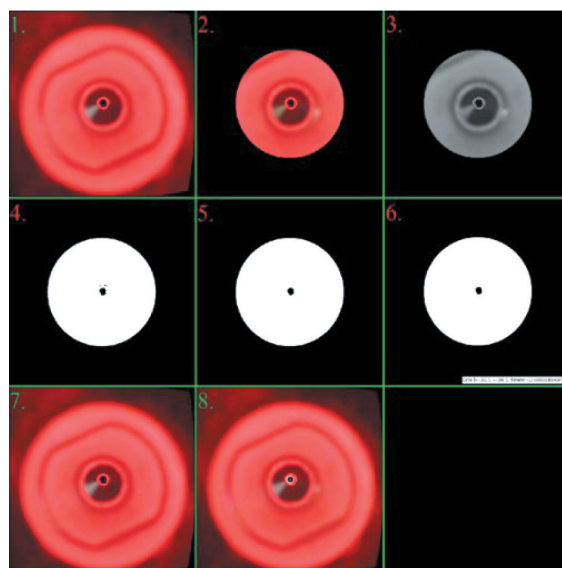
Algorytmy używane przez kamerę główną różnią się od tych stosowanych przez dolną, ponieważ służą one głównie do wykrywania otworów referencyjnych na taśmie z elementami czy płytkach PCB. Podczas automatycznej pracy manipulator przesuwa górną kamerę nad pozycję, w której powinien znajdować się następny element do podniesienia i sprawdza, gdzie i czy znajdują się otwory referencyjne taśmy. Dzięki temu nie jest wymagane bardzo dokładne kalibrowanie długich taśm (przy których ewentualny błąd byłby bardziej zauważalny), ponieważ przy każdym podniesieniu oprogramowanie sprawdza i wykonuje ewentualną poprawkę pozycji.

W teorii algorytm wykrywania otworów referencyjnych może przypominać ten służący do wykrywania końcówki głowicy. Jednak z powodu różnych ustawień kamer oraz różnych warunków oświetlenia tych elementów algorytmy te musiały być dobrane w inny sposób. Ostatnim już sposobem wykrywania elementów jest algorytm służący do znajdowania otworów referencyjnych na płytkach PCB. Używany jest on do

automatycznego ustalania pozycji płytki w polu roboczym. Oszczędza to pracy użytkownikowi, ponieważ elementy muszą być ułożone z bardzo dużą dokładnością, więc ręczne ustalanie pozycji przy każdym rozpoczęciu pracy byłoby uciążliwe.

7. OPIS PRACY MANIPULATORA

Po skonfigurowaniu manipulatora można przystąpić do wykonania pierwszej automatycznej pracy maszyny. Po włączeniu maszyny należy zacząć od konfiguracji błędu nieosiowości. Praca algorytmu wykrywającego końcówkę głowicy pokazana jest na rysunku 12. Jak można na nim zauważyć, obraz najpierw ograniczany jest okręgiem, a następnie jego kodowanie zmieniane jest na odcienie szarości. W kolejnym etapie wykrywane są piksele, które znajdują się powyżej wybranej intensywności skali szarości. Obraz czyszczony jest z zakłóceń i w etapie 6 znajduje się okrąg. Etap 8 pokazuje, że algorytm zadziałał poprawnie. Instrukcje te wykonywane są dla 6 różnych obrotów głowicy, aby wykryć błąd przy aktualnej konfiguracji osi głowicy.

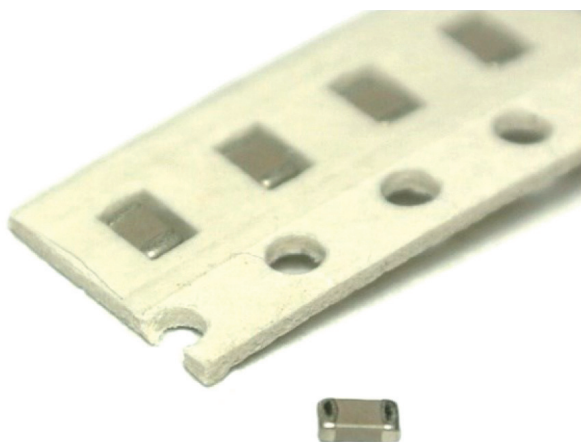


Rys. 12. Praca algorytmu wykrywającego końcówkę głowicy manipulatora (numeracja zgodna z algorytmem przedstawionym w części 6 artykułu)

Kolejnym krokiem jest przygotowanie pliku CAD wykonanego w jednym z wielu oprogramowań do tego przeznaczonych i zapisaniu go w formacie CSV lub TXT. Oba te formaty zawierają spis wszystkich danych potrzebnych do ułożenia elementów, tj. pozy-

cja, rotacja czy rodzaj elementu. Występuje wiele rodzajów komponentów układanych na płytkach.

Na potrzeby tego artykułu opisana zostanie praca układania jednego elementu, a dokładnie kondensatora 0805 przedstawionego na rysunku 13.



Rys. 13. Kondensator 0805 [12]

Następnie należy wpisać do oprogramowania OpenPnP pozycję płytki PCB w polu roboczym manipulatora, ponieważ koordynaty znajdujące się w plikach liczone są od jednego z rogów płytki. Na płytkach PCB znajduje się zazwyczaj od trzech do czterech otworów referencyjnych. Po sprawdzeniu pierwszego manipulator porusza się więc nad drugi, a następnie trzeci (potem czwarty, jeżeli istnieje) i cały proces jest powtarzany, aż obliczona jest dokładna pozycja i obrót płytki PCB w polu roboczym maszyny.

7.1. Automatyczne układanie

Kolejną czynnością jest już włączenie automatycznej pracy manipulatora. Po rozpoczęciu procesu układania manipulator podjeżdża nad podajnik z elementami. Oprogramowanie wykonuje algorytm sprawdzenia pozycji otworów referencyjnych taśm, którego końcowym wynikiem jest znalezienie wszystkich widocznych otworów referencyjnych. Zamontowanie kilku taśm obok siebie nie sprawia problemu, ponieważ zastosowana instrukcja przycina obraz tak, aby widoczny był jedynie potrzebny jego fragment.

Warto również wspomnieć o pracy oświetlenia. Diody LED nie powinny być zaświecone przez cały czas pracy manipulatora, a jedynie przez krótką chwilę, kiedy obraz jest pobierany z kamery. Umożliwiają to dwa zastosowane skrypty [13] napisane w języku JavaScript. Pierwszy z nich uruchamiany jest podczas

inicjalizacji instrukcji związanych z odczytaniem widoku kamery. Sprawdza on, które z urządzeń ma wykonać tę pracę, i włącza przeznaczone dla niego oświetlenie, jednocześnie wyłączając diody znajdujące się po przeciwnej stronie.

Działanie skryptu drugiego jest prostsze od poprzednika. Początkowo również zapisywane jest odwołanie do włączników oświetlenia, a następnie są one wyłączane. Nie jest potrzebne zastosowanie instrukcji warunkowej, aby sprawdzać, które z diod były włączone, ponieważ wyłączenie niezapalonego oświetlenia nie wywołuje żadnej reakcji. Cały proces pobrania obrazu z kamery trwa niecałą sekundę.

Po sprawdzeniu pozycji otworów referencyjnych taśm z elementami oprogramowania oblicza ewentualną poprawkę pozycji, jeżeli znalezione okręgi różnią się od tego, co było zapisane w pamięci. Następnie wykonywany jest ruch osiami X oraz Y pozycjonujący głowicę dokładnie nad pobieranym elementem. Po jego zakończeniu oś Z porusza się tak, aby końcówka głowicy dotykała elementu. W tym momencie włączana jest pompka tworząca podciśnienie i część jest podnoszona. Dalej manipulator przesuwa podniesiony komponent tak, aby znajdował się on w środku obrazu dolnej kamery i jednocześnie na wysokości płytki PCB, na której ma zostać ułożony. Podczas wykonywania tego ruchu jest on też wstępnie obracany. Kiedy element znajduje się już w zamierzonej pozycji, uruchamiany jest kolejny algorytm przetwarzania obrazu. W tym przypadku obraz również jest ograniczany, tym razem okręgiem, aby zminimalizować liczbę elementów mogących utrudniać wykrycie komponentu.

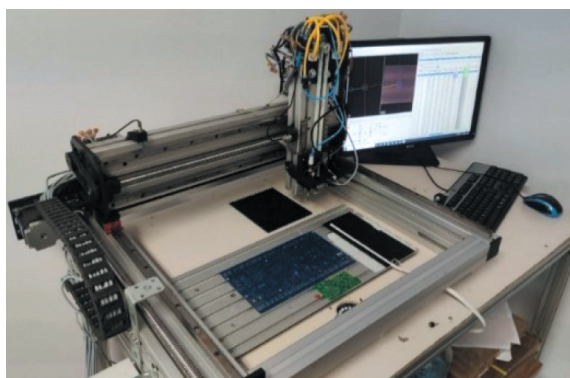
Zastosowane oświetlenie ma kolor czerwony, co pozwala na łatwe odróżnienie elementu od reszty konstrukcji, co wykonywane jest podczas etapu trzeciego i czwartego. Następnie obraz czyszczony jest z zakłóceń i pozostaje jedynie biały odcisk elementu na czarnym tle. Dzięki temu wykryty obrót oraz wymiary komponentu są prawie idealne. Jeżeli wykryty kąt obrotu elementu jest inny niż ten zamierzony, głowica obraca się o wyliczony błąd. Algorytm wykonuje swoją pracę jeszcze raz, jeżeli poprawka wynosi więcej niż 5° . Oblicza on również odsunięcie elementu od środka obrazu. Głowica nie zawsze podnosi element dokładnie w centrum jego powierzchni, co może powodować nieduże przesunięcia przy układaniu.

Kiedy zostaną obliczone wszystkie poprawki, wykonywana jest sekwencja końcowa. Oś Z podnosi się na pozycję 0, osie X i Y wykonują ruch, pozycjonując

kondensator dokładnie nad miejscem ułożenia. Następnie głowica zniża się tak, aby komponent dotykał płytki oraz był dociśnięty. Końcówka głowicy jest skonstruowana z użyciem sprężyny. Pozwala to na docisk elementu do płytki PCB. Kiedy ruchy zostaną zakończone oraz minie czas docisku (został on ustawiony na 1000 ms), przełączany jest elektrozawór. Kolejnym wykonywanym ruchem jest podniesienie głowicy do pozycji 0. Ruchy te wykonywane są równoległe dla obu głowic. W momencie kiedy oba elementy są ułożone, algorytm pracy zaczyna od początku, tj. kamera ustawiana jest nad taśmą z elementami itd. Jeżeli ułożony został ostatni element, manipulator przesuwa się na pozycję parkingową i automatyczna praca zostaje zakończona.

7.2. Podsumowanie

Czas wykonywania pracy jest zależny od złożoności projektu płytki PCB oraz zastosowanej prędkości silników. Im więcej elementów do ułożenia, tym dłużej trwa proces układania. Zwiększenie prędkości spowoduje szybsze wykonywanie pracy przy zmniejszeniu dokładności. Należy również pamiętać o zapewnieniu odpowiedniej liczby komponentów, aby podczas wykonywania pracy ich nie zabrakło. Jeżeli tak się stanie, algorytm zostanie zatrzymany i oprogramowanie będzie wymagało interwencji użytkownika. Po zakończeniu pracy manipulator znajduje się w pozycji parkingowej pokazanej na rysunku 14.



Rys. 14. Manipulator znajdujący się w pozycji parkingowej

8. WNIOSKI

Budowa manipulatora Pick and Place jest skomplikowana głównie z powodu wymaganej precyzji

działania. Podczas procesu projektowania i konstrukcji maszyny wyciągnięto wnioski:

- Przy zastosowaniu sterownika Mega2560 wraz z nakładką RAMPS 1.6 nie ma możliwości bezpośredniego sterowania diodami LED. Wymagane było zastosowanie dodatkowego sterownika Arduino Nano, co spowodowało wzrost kosztów konstrukcji.
- Konfiguracja algorytmów przetwarzających obraz wymaga częstych zmian oświetlenia, aby udało się dobrać najbardziej odpowiednie etapy jego pracy.
- Zastosowanie gotowej ramy zmniejszyło koszty konstrukcji maszyny, jednak posiada ona napęd śrubowy oraz jest stosunkowo ciężka, powoduje to, że prędkość poruszania się osi jest niewielka, co wydłuża czas automatycznego układania.
- Montaż manipulatora do stołu z kołami ułatwia zmianę umiejscowienia maszyny, jednak powoduje to, iż łatwiej można spowodować przesunięcie elementów ułożonych w polu roboczym, np. przez przypadkowe uderzenie w konstrukcję.
- W tym artykule została przedstawiona jedynie konfiguracja dla jednego elementu SMD, tj. kondensatora 0805. Do pełnego automatycznego ułożenia całego projektu płytki PCB należałoby wprowadzić ustawienia każdego układanego komponentu.

Konstrukcja manipulatora umożliwia automatyczne układanie elementów, jednak aby uzyskać zmniejszenie czasu pracy maszyny, można wykonać kilka dodatkowych zabiegów. Pierwszym z nich jest dodanie i skonfigurowanie drugiej głowicy. Konstrukcja mechaniczna i elementy elektroniczne manipulatora zostały do tego przystosowane, co zdecydowanie ułatwia ten proces. Kolejnym dodatkiem może być skonstruowanie automatycznych podajników elementów SMD. Kiedy są one zamontowane do manipulatora, możliwe jest podanie większej liczby elementów jednocześnie, co zmniejsza częstotliwość zmieniania pustych taśm. Sterowanie nimi odbywałoby się przez zastosowanie odpowiednich skryptów.

Wydajność manipulatora zależy zarówno od ułożenia płytek PCB na przestrzeni roboczej, jak i ułożenia podajników elementów SMD. Im większą odległość muszą pokonać głowice, tym mniejsza jest wydajność, jednak według szacunków manipulator jest w stanie ułożyć około 180–200 części na godzinę.

Literatura

- [1] Andurkar G.K., Borkar V.: *Development of Pick and Place Robot for Industrial Applications*. International Research Journal of Engineering and Technology (IRJET), 4, 9, 2017.
- [2] *Sneak Preview: DIY SMT Pick & Place Machine with OpenPnP*. <https://mcuoneclipse.com/2018/05/05/sneak-preview-diy-smt-pick-place-machine-with-openpnp/> [15.12.2020].
- [3] *Pick-and-Place Workflow using Stateflow for Matlab*. <https://www.mathworks.com/help/robotics/examples/pick-and-place-workflow-using-stateflow.html> [15.12.2020].
- [4] *LitePlacer standard kit*. https://www.liteplacer.com/shop20/index.php?route=product/product&product_id=64 [18.12.2020].
- [5] *The History of Industrial Robots, From Single Taskmaster to Self-Teacher*. <https://redshift.autodesk.com/history-of-industrial-robots/> [15.12.2020].
- [6] *JUKI RS-1R SMT Pick and Place Machine*. https://www.smt11.com/product/Pick-and-Place-Machine/JUKI-RS-1R-SMT-Pick-and-Place-Machine-78914.html#PRODUCT_CHARACTERISTICS [21.12.2020].
- [7] *BARR group: Introduction to Pulse Width Modulation (PWM)*. <https://barrgroup.com/Embedded-Systems/How-To/PWM-Pulse-Width-Modulation> [23.12.2020].
- [8] *Coordinate System*. <https://github.com/openpnp/openpnp/wiki/User-Manual#coordinate-system> [29.12.2020].
- [9] Evans M., Noble J., Hochenbaum J.: *Arduino w akcji*. Helion, Gliwice 2014: 124–125.
- [10] *Setup and Calibration*. <https://github.com/openpnp/openpnp/wiki/Setup-and-Calibration> [3.01.2021].
- [11] *Nozzle Tip Calibration Setup*. <https://github.com/openpnp/openpnp/wiki/Nozzle-Tip-Calibration-Setup> [6.01.2021].
- [12] *Kondensator 0805*. http://www.sm-elektronik.pl/galerie/1/10uf-6-3v-x7r-smd-0805-kondens_5882.jpg [9.01.2021].
- [13] *Scripting*. <https://github.com/openpnp/openpnp/wiki/Scripting> [10.01.2021].

dr inż. KRYSZTOF SZOPA

inż. PAWEŁ KARELUS

Wydział Inżynierii Mechanicznej i Robotyki

AGH Akademia Górniczo-Hutnicza

im. Stanisława Staszica w Krakowie

al. A. Mickiewicza 30, 30-059 Kraków

kszopa@agh.edu.pl

pkarelus@student.agh.edu.pl