

Kierunek: Informatyka algorytmiczna (INF-PPT)

Specjalność: -

PRACA DYPLOMOWA
INŻYNIERSKA

Prototyp maszyny vendingowej

Piotr Gałwiazek

Opiekun pracy
Dr inż. Przemysław Błaśkiewicz

Słowa kluczowe: Arduino, sterowanie, automat sprzedażowy

Streszczenie pracy w języku polskim

Celem pracy „Prototyp maszyny vendingowej” jest stworzenie prototypu automatu vendingowego.

Głównym problemem jest zaprogramowanie komponentów użytych w projekcie, takich jak: Wrzutnik monet CH-923 obsługujący monety pięcio-, dwu-, jedno-złotowe, płyta sterująca Arduino Mega 2560, silniki krokowe 28BYJ-48 ze sterownikiem ULN 2003, ekran LCD z konwerterem I2C.

Kolejną trudnością jest połączenie tych elementów w taki sposób, aby były ze sobą kompatybilne.

W pracy zawarta jest historia branży vendingowej w Polsce i na świecie.

Przedstawiam również porównanie ze sobą mechanizmów wydawania produktu oraz opis komponentów, które obecnie znajdują się na rynku vendingu.

Kolejnym etapem jest zestawienie najbardziej popularnych metod płatności występujących w dzisiejszych czasach w maszynach vendingowych.

Praca zawiera również opis trybu serwisowego, który jest obsługiwany przez osobę serwisującą.

Następną fazą projektu jest specyfikacja komponentów użytych w tej konkretnej koncepcji.

Wstawione rysunki przedstawiają schematy połączeń komponentów stworzone za pomocą programu Fritzing, diagramy UML działania automatu oraz implementację kodu napisanego w środowisku Arduino. Szczegółowo opisałem również programowanie wrzutnika monet.

Czwarty etap pracy zawiera testy i wnioski. Pokazany zostaje cały przebieg testowania automatu – od włączenia maszyny do zasilania, przez wrzucanie zaprogramowanych monet, aż do zakupu wybranego produktu. We wnioskach zamieszczam opis możliwego rozwinięcia projektu w różnych kierunkach oraz podsumowania działania projektu w fazie finalnej.

Streszczenie pracy w języku angielskim

The aim of the study "Prototype of a vending machine" is to create a prototype of a vending machine.

The main problem is programming the components used in the project, such as: CH-923 coin acceptor supporting coins several values, Arduino Mega 2560 control board, 28BYJ-48 stepper motors with ULN 2003 driver, LCD screen with I2C converter.

Another difficulty is combining these elements in such a way that they are compatible with each other.

The work contains the history of the vending industry in Poland and in the world.

I also present a comparison of the product dispensing mechanisms and a description of the components that are currently on the vending market.

The next step is to compile the most popular payment methods used in vending machines nowadays.

The work also includes a description of the service mode, which is operated by a service person.

The next phase of the project is the specification of the components used in this particular concept.

The inserted drawings show components connection diagrams created with the Fritzing program, UML diagrams of the machine operations and the implementation of the code written in the Arduino environment. I also described the programming of the coin acceptor in detail.

The fourth stage of work includes tests and conclusions. The entire course of testing the machine is shown - from turning on the machine on to the power supply, through inserting programmed coins, to purchasing the selected product. In the conclusions, I include a description of the possible development of the project in various directions and a summary of the project's operations in the final stage.

Spis treści

1. Wstęp	5
1.1. Cel i zakres pracy	5
1.2. Przedstawienie komponentów i opis problemu	5
1.3. Historia automatów vendingowych	5
2. Maszyna Vendingowa.....	7
2.1. Wydawanie produktu	7
2.2. Tryb serwisowy.....	10
2.3. Wydawanie produktu z problemami	10
2.4. Płatności.....	11
3. Realizacja oraz specyfikacja komponentów użytych w projekcie.....	14
3.1. Specyfikacja komponentów	14
3.2. Schemat połączeń komponentów.....	18
3.3. Diagram UML.....	20
3.4. Implementacja.....	21
3.4.1. Biblioteki.....	21
3.4.2. Główna pętla programu.....	21
3.4.3. Tryb serwisowy	22
3.4.4. Wybór produktu	23
3.4.5. Programowanie wrzutnika monet.....	24
3.4.6. Silniki krokowe	26
4. Testy i wnioski.....	27
4.1. Testowanie automatu	27
4.2. Wnioski.....	30

1. Wstęp

1.1. Cel i zakres pracy

Celem pracy jest stworzenie prototypu małej maszyny vendingowej. Podstawową funkcjonalnością systemu jest możliwość zakupu jednego z czterech oferowanych produktów za pomocą nominałów funkcjonujących obecnie w obiegu.

Maszyna oferuje również przyjazny dla użytkownika tryb serwisowy. Tryb ten jest przeznaczony dla serwisanta, który będzie chciał uzupełnić produkty bądź sprawdzić stan bieżący monet znajdujących się w automacie.

1.2. Przedstawienie komponentów i opis problemu

Głównym problemem poruszonym w pracy jest zaprogramowanie komponentów, które zostały użyte w projekcie. Na urządzenie sterujące całym układem wybrałem płytkę sterującą Arduino MEGA 2560. To ona będzie odpowiadać w projekcie za sterowanie mechanizmem wrzutowym monet, silnikami oraz ekranem. Kolejnym komponentem jest wrzutnik monet, który został zaprogramowany tak, aby odpowiadał za filtrowanie monet w ten sposób, aby akceptowały tylko monety pięcio, dwu oraz jednozłotowe.

Za wyświetlanie informacji dla użytkownika będzie odpowiadał ekran LCD o wymiarach 8 cm x 3,6 cm. Mechanizm wypadania produktu będzie oparty o obrót metalowej spirali o 360 stopni zgodnie z ruchem wskazówek zegara. Produkty będą znajdować się w spirali, co podczas obrotu tej sprężyny sprawia, że spadają z półki. Do obracania sprężyn użyję silników krokowych z przekładnią 28BYJ-48 o napięciu 5V, natężeniu 0,1 A i momencie obrotowym 0,03 Nm ze sterownikiem ULN2003. Za zasilanie wrzutnika monet odpowiada zasilacz do taśm ledowych o napięciu 12V. Do zasilenia płytki sterującej oraz silników użyję ładowarki o napięciu 5V. Dla estetyki wykonania, ładowarki zasilające będą podłączone do przedłużacza, który będzie znajdował się wewnątrz automatu. Finalnie z maszyny będzie wychodził wyłącznie jeden kabel zasilający o napięciu 230V.

1.3. Historia automatów vendingowych

Automaty vendingowe stają się coraz bardziej popularne w Polsce i na rynku światowym, ze względu na swoją prostotę i koszty produkcji w porównaniu do sklepów stacjonarnych. Ponadto maszyny te są w pełni zautomatyzowane, co sprawia, że rozwiązanie wyklucza zatrudnianie ludzi jako sprzedawców.

Chociaż mogłoby się wydawać, że automaty vendingowe zostały wymyślone w ciągu ostatnich kilkunastu lat to tak naprawdę ich historia sięga lat 80 XIX wieku.

Zlokalizowane były w pobliżu budynków poczty, czy przy dworcach kolejowych. Dzięki temu podróżni mogli szybko i bez potrzeby chodzenia po mieście wysłać pozdrowienia czy ważny list.

A co z maszynami sprzedającymi napoje, które są jednymi z najpopularniejszych w gałęzi vendingu? Te pojawiły się niedługo później, a kawy z automatu można było się napić dopiero w 1947 roku dzięki Amerykanom.

Vending w Polsce pojawił się stosunkowo późno bo dopiero po denominacji w 1995 roku, kiedy to w obiegu pojawiły się nowe monety. Powodem tej sytuacji były również przeszkody prawne w okresie PRL oraz brak dostępu do nowoczesnego sprzętu.

Rynek vendingu jest bardzo popularny na całym świecie. Krajami przodującymi w tej branży jest Japonia oraz Chiny. Znajdują się tam już automaty oferujące klientom ciepłą pizzę, poddaną obróbce termicznej bezpośrednio w automacie. Branża również rozwinęła się pod względem oferowanych sposobów płatności. Można spotkać automaty oferujące płatność za pomocą kart kredytowych, telefonów czy też poprzez skanowanie kodu QR bezpośrednio z maszyny i dokonaniu zapłaty poprzez aplikację oferowaną przez dostawcę usług płatniczych. Według Polskiego Stowarzyszenia Vendingu rynek branży vendingowej notuje coraz lepsze wyniki. W roku 2019 jego wartość w Polsce szacuje się na około 900 mln zł. [1]



Rys. 1.1 Automat sprzedający pocztówki (Japonia, 1904) [1]

2. Maszyna Vendingowa

2.1. Wydawanie produktu

W obecnych maszynach vendingowych mamy do dyspozycji wiele mechanizmów obsługujących wydawanie produktów. Oto kilka z nich:[2]

- Winda z pojemnikiem

Automat jest wyposażony w windę, która pomaga bezpiecznie przetransportować wybrany przez klienta produkt do komory odbiorczej. Winda w budowie przypomina pojemnik, który za pomocą silnika przesuwany jest w stronę wybranego produktu. Kolejny silnik wypycha produkt za pomocą plastikowego mechanizmu bezpośrednio do pojemnika. Następnie pojemnik kieruje się w stronę komory odbiorczej, aby następnie przechylić i zrzucić produkt.

- Winda z chwytakiem

Automat jest wyposażony w chwytak, który „zabiera” produkt z odpowiedniej półki i transportuje w kierunku komory odbiorczej. Następnie ramiona chwytaka otwierają się, co sprawia, że produkt spada do komory. Chwytak w maszynie vendingowej przypomina chwytak do drewna.

- System spiralowy

Automat jest wyposażony w system spiralowy. Spirala znajduje się na półce poziomo i jej początek skierowany jest w kierunku przodu automatu. Na drugim końcu sprężyny (tylna ściana automatu) znajduje się silnik, który jest przytwierdzony do spirali za pomocą kołowrotka. Silnik wykonuje obrót o 360 stopni, co sprawia, że pierwszy produkt wypada poza półkę, na której się znajduje i łąduje w komorze odbiorczej i jest gotowy do odbioru przez klienta. Obracanie się sprężyny sprawia, że wszystkie znajdujące się na niej produkty przesuwają się o jedno miejsce do przodu.

- System talerzowy

Automat wyposażony jest w system talerzowy służący do wydawania produktów. W takim automacie znajduje się walec, którego wysokość jest podzielona na x półek. Te półki to „talerze”. Talerze, patrząc na nie od góry dzielą się na wycinki koła o różnym kącie wewnętrznym. System ten jest stosowany w automatach, w których produkty są delikatne, a upadek z wysokości mógłby je uszkodzić. W tym systemie omijamy krok, w którym produkt spada. Walec obraca się, a klient ma bezpośredni dostęp do produktu, który wybrał, ponieważ odblokowuje się „klapka”.



Rys. 2.1 Automat vendingowy wykorzystujący system spiralowy



Rys. 2.2 Automat vendingowy z systemem windy z pojemnikiem [2]



Rys. 2.3 Automat vendingowy przedstawiający mechanizm chwytakowy bez windy [3]



Rys. 2.4 Automat vendingowy przedstawiający system talerzowy [4]

Wśród wymienionych systemów znajduje się ten użyty przeze mnie w pracy inżynierskiej. Jest to system spiralowy bez windy, jedno z najpopularniejszych rozwiązań w tego typu automatach. Nie posiada on windy, ponieważ produkty, które będą umieszczone w automacie nie są delikatne i nie uszkodzą się w przypadku upadku z najwyższej półki.

2.2. Tryb serwisowy

Zbudowany automat vendingowy zawiera tryb serwisowy, który ma na celu stworzenie komunikacji między automatem, a osobą serwisującą. Aby uruchomić tryb serwisowy, serwisant otwiera automat, następnie łączy się z płytką sterującą Arduino za pomocą kabla USB. Po podłączeniu laptopa z oprogramowaniem Arduino do płyty sterującej na ekranie pojawia się napis: „Tryb serwisowy”. Serwisant po wpisaniu hasła jest w stanie sprawdzić ile dokładnie produktów zostało sprzedanych na poszczególnych sprężynach oraz sumę pieniędzy, jaka została zebrana.

2.3. Wydawanie produktu z problemami

System wydawania produktu został wyposażony w zabezpieczenia. Jednym z problemów, jakie możemy napotkać w trakcie zakupu jest sytuacja, w której to klient próbuje kupić produkt, którego nie ma na półce. Sytuację rozwiązuje zabezpieczenie napisane w kodzie, które

uniemożliwia zakup produktu, którego fizycznie nie ma na stanie. Dla klienta wygląda to w taki sposób, że po wrzuceniu monet, na ekranie brak jest litery oznaczającej produkt, którego stan wynosi 0 oraz przycisk nie reaguje na naciśnięcie. Klient może wtedy wybrać inny produkt. W przypadku, gdy w automacie brak jest jakiegokolwiek produktu, ekran wyświetla informację „Brak produktów” oraz „Obsługa klienta tel: 511 *** 834”.

2.4. Płatności

Dzisiejsze automaty vendingowe oferują nam rozmaity wachlarz płatności. Wraz ze wzrostem popularności kart płatniczych oraz rozwojem technologii na rynku pojawiła się różnorodność systemów płatniczych. Wśród nich można wyróżnić:

- Płatność monetami

Jednym z podstawowych i najczęściej używanym rozwiązaniem jest wrzutnik monet, gdyż w większości automatów oferowane produkty posiadają niską cenę, co sprawia, że jest to najlepsza i najbardziej skuteczna metoda. Urządzenie to zaprogramowane jest tak, aby przyjmowało żądane przez nas monety. Niektóre z nich oferują również możliwość wydawania reszty. (Rys. 2.5)

- Płatność banknotami

System ten pozwala na zakup produktów za pomocą banknotów. Używa się go najczęściej jako dodatek do wspomnianego wyżej wrzutnika monet. Akceptory banknotów nie wydają reszty w postaci banknotów, lecz współpracują z wrzutnikami monet, które wydają resztę za pomocą monet. Systemy te nie są tak popularne jak wrzutniki monet, ponieważ używa się ich do mniej popularnych, droższych produktów oferowanych przez automat. (Rys.2.6)

- Płatności zbliżeniowe

Najpopularniejszym w obecnych czasach stał się system płatności zbliżeniowych. System obsługiwany jest za pomocą terminala płatniczego zamontowanego na froncie automatu. Jest to wygodna alternatywa dla systemu płatności monetami, ponieważ klient nie musi posiadać banknotów lub monet, a płatność odbywa się przez przyłożenie karty płatniczej, bądź telefonu do terminala płatniczego. (Rys.2.7)

- Płatności za pomocą kodu QR

Kolejnym systemem, w Polsce na tą chwilę jeszcze rzadko używanym jest płatność za pomocą kodu alfanumerycznego znajdującego się w większości przypadków na frontowej ścianie automatu. Klient chcący dokonać zakupu wybranego wcześniej produktu skanuje kod QR za pomocą skanera w smartfonie, co przenosi go do strony internetowej operatora płatności, gdzie po potwierdzeniu przelewa określoną ilość środków w celu dokonania zakupu. (Rys.2.8)

- Płatności audio

Jednym z najnowszych systemów umożliwiającym dokonanie zakupu w automacie vendingowym okazuje się system płatności dźwiękowych. Rozwiązanie to oferuje chińska

firma Alipay. Po wybraniu produktu w automacie, płacimy aplikacją, która token płatności zapisuje za pomocą pliku dźwiękowego. Po przyłożeniu telefonu do głośnika maszyny vendingowej i odtworzeniu nagrania transakcja zostaje zaakceptowana, a my możemy odebrać produkt z komory odbiorczej automatu. Rozwiązanie to na tą chwilę nie spotykane w naszym kraju okazuje się dobrym i tanim, ponieważ automat vendingowy musi posiadać tylko głośnik. Nie potrzebny okazuje się urządzenie do wrzucania monet, akceptor banknotów, czy też terminal płatniczy, których to cena okazuje się znacząco wyższa. (Rys.2.9)



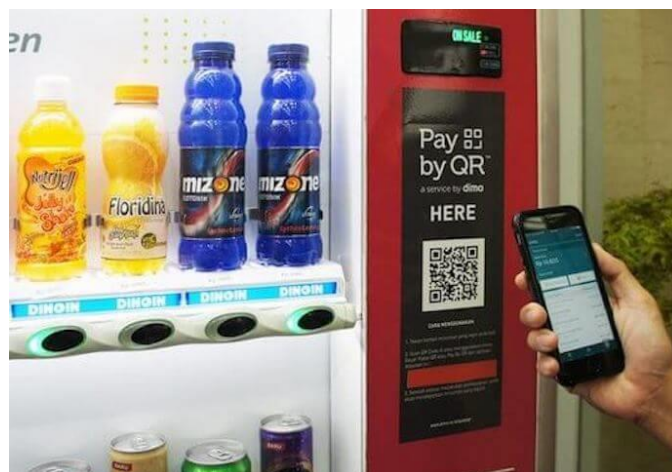
Rys. 2.5 Wrzutnik monet obsługujący system wydawania reszty [5]



Rys. 2.6 Akceptor banknotów [5]



Rys. 2.7 Terminal płatniczy [5]



Rys. 2.8 Płatność za pomocą kodu QR [6]



Rys. 2.9 Płatność audio [7]

3. Realizacja oraz specyfikacja komponentów użytych w projekcie

3.1. Specyfikacja komponentów

- Płytką sterującą Arduino Mega 2560 Rev3 [8]

Arduino Mega jest jedną z najbogatszych wersji modułów Arduino. Płytką zawiera mikrokontroler ATmega2560, wyposażony w 54 cyfrowych wejść/wyjść z czego 15 można wykorzystać jako wyjścia PWM (np. do sterowania silnikami) oraz 16 analogowych wejść. Układ taktowany jest sygnałem zegarowym o częstotliwości 16 MHz, posiada 256 kB pamięci programu Flash oraz 8 kB pamięci operacyjnej SRAM.



Rys. 3.1 Płytką sterującą Arduino Mega 2560 Rev3 [8]

Specyfikacja produktu:

- Napięcie zasilania: 7 V do 12 V
- Mikrokontroler: ATmega 2560
 - Maksymalna częstotliwość zegara: 16 MHz
 - Pamięć SRAM: 8 kB
 - Pamięć Flash: 256 kB (8 kB zarezerwowane dla bootloadera)
 - Pamięć EEPROM: 4 kB
- Piny I/O: 54
- kanały PWM: 15
- Ilość wejść analogowych: 16 (kanały przetwornika A/C o rozdzielczości 10 bitów)
- Interfejsy szeregowo: 4xUART, SPI, I2C
- Zewnętrzne przerwania
- Podłączona dioda LED do pinu 13
- Gniazdo USB A do programowania
- Złącze DC 5,5 x 2,1 mm do zasilania płytki

- Wrzutnik monet (Multi coins Acceptor BLEE 6) [9]

Akceptor umożliwia stworzenie urządzenia, którego działanie będzie uzależnione od wrzucenia monety (poboru opłaty). Wrzutnik obsługuje sześć typów monet, których rodzaje można zaprogramować przy użyciu przycisków i wyświetlacza. Czujnik zastosowany w urządzeniu identyfikuje monety wykorzystując ich średnicę, grubość oraz czas opadania. Do zaprogramowania odpowiednich monet służy wyświetlacz z przyciskami z boku akceptora.[9]



Rys. 3.2 Wrzutnik monet (Multi coins Acceptor BLEE 6) [9]

Specyfikacja produktu:

- Obsługuje 6 rodzajów monet
- Obsługa wyjścia wielu sygnałów
- Stosowanie średnicy monet: 15 - 29 mm
- Stosowanie grubości monet: 1,8 – 3,0 mm
- Ciśnienie atmosferyczne: 90Kpa – 106Kpa
- Napięcie robocze: DC12V +/- 10%
- Prąd roboczy: 50mA
- Rozmiar produktu: 64 x 124 x 143 mm

- Wyświetlacz LCD 2x16 z magistralą LCM1602 I2C [10]

Popularny alfanumeryczny wyświetlacz LCD, zasilany napięciem 5 V. Charakteryzuje się prostą obsługą, wysoką dostępnością oraz licznym wsparciem dla wielu mikrokontrolerów. Dodatkowo do ekranu dołączony zostaje moduł LM1602 umożliwiający podłączenie wyświetlacza do mikrokontrolera poprzez magistralę I2C. Dzięki takiemu połączeniu zamiast 6 linii (D4, D5, D6, D7, E, RS) użyjemy tylko dwóch. Pozwoli to zaoszczędzić cenne wyprowadzenia w Arduino Mega 2560. Dzięki magistrali nie będzie potrzebny również potencjometr regulacji kontrastu, gdyż znajduje się on już w module LM1602. Daje nam to kontrolę nad podświetleniem wyświetlacza – można je w dowolnym momencie włączać lub wyłączać korzystając z biblioteki LiquidCrystal I2C.



Rys. 3.3 Wyświetlacz LCD 2x16 [10]

Specyfikacja produktu:

- Typ: wyświetlacz LCD alfanumeryczny
- Napięcie zasilania: 5 V
- Sterownik zgodny z HD44780
- Podświetlanie: białe znaki
- Niebieskie tło
- Zakres temperatury pracy: od -20°C do 70°C
- Wymiary jednego znaku: 2,45 x 5 mm
- Wymiary modułu: 80 x 36 mm

- 4 silniki krokowe 28BYJ-48 z przekładnią [11]

Jeden z najpopularniejszych silników krokowych ze zintegrowaną przekładnią 28BYJ-48. Moduł ze sterownikiem ULN2003 umożliwia kontrolę kierunku obrotów i prędkości 4-fazowych silników krokowych, dedykowany dla Arduino.



Rys. 3.4 Silnik krokowy 28BYJ-48 ze sterownikiem ULN2003 V2 [11]

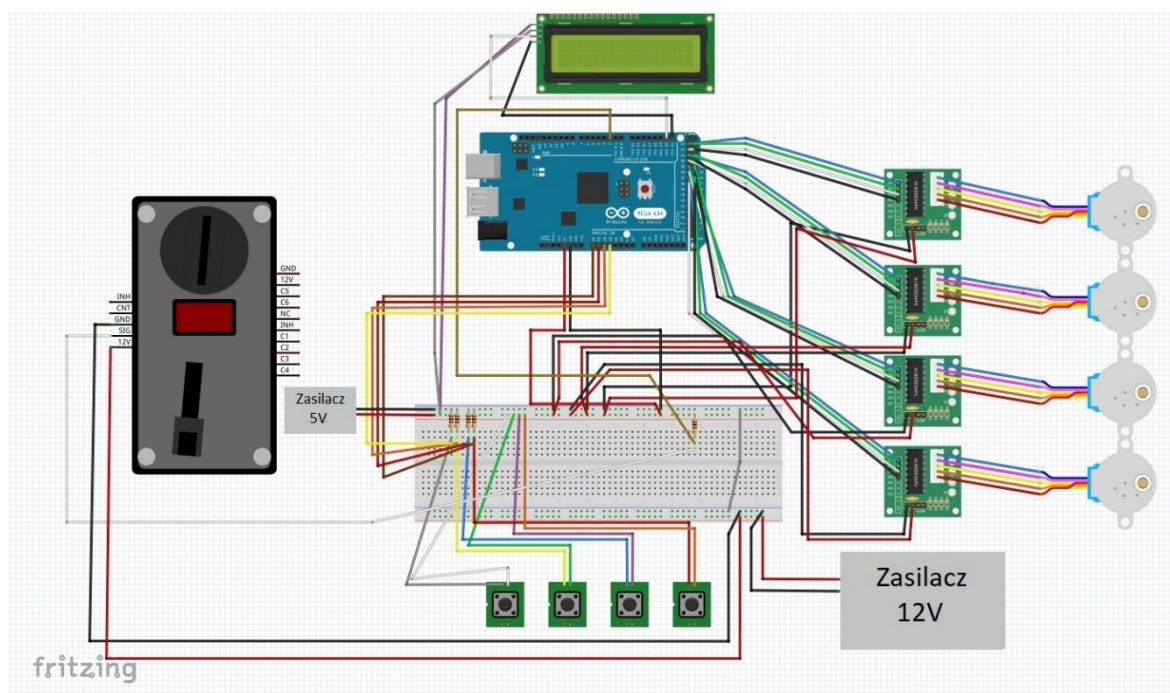
Specyfikacja silnika:

- moment trzymający na wyjściu przekładni: 0,3 kg*cm (0,03 Nm)
- przełożenie przekładni: 64:1
- napięcie znamionowe: 5V
- pobór prądu na cewkę: 100 mA
- rezystancja cewki: 50Ω
- wyprowadzenia: pięć przewodów
- średnica wału: 5mm ze ścieżkami
- masa: 35g
- odległość pomiędzy otworami montażowymi: 35mm
- wymiary: $\phi 28 \times 19$ mm (bez wału)
- długość przewodów: ok 24cm

Specyfikacja sterownika ULN2003 V2:

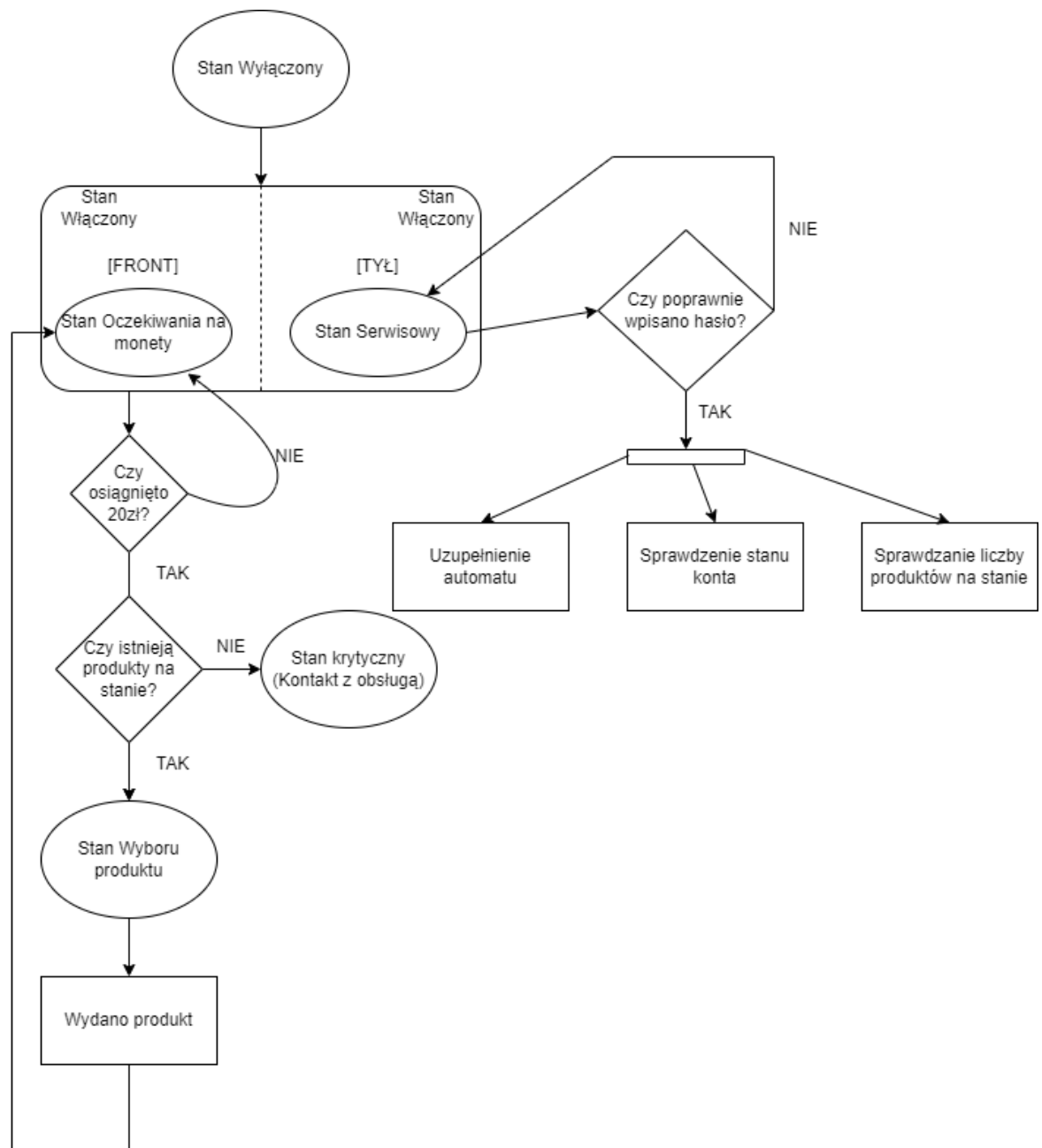
- zasilanie napięciem: 5V DC
- napięcie zasilania: do 12V
- napięcie zasilania części logicznej: 5 V
- maksymalny prąd wyjściowy na kanał: 500 mA
- prąd zasilania układu: 10mA
- maksymalna moc strat: 5W
- maksymalny prąd na kanał: 500 mA
- sygnalizacja stanu pracy: 4x LED
- złącze silnika: XH-05
- wymiary: 40,5x23mm
- cztery piny wejściowe: IN1, IN2, IN3, IN4

3.2. Schemat połączeń komponentów



Rys. 3.5 Schemat połączeń komponentów w programie FRITZING

3.3. Diagram UML



Rys. 3.7 Diagram UML

Diagram UML przedstawia schemat możliwych stanów automatu. Za działanie każdego stanu odpowiadają odpowiednie funkcje zamieszczone w kodzie.

- Stan wyłączony - automat nie jest podłączony do źródła zasilania
- Stan włączony - dzieli się na dwa stany. Front przedstawia stan oczekiwania na monety. Stanem współbieżnym jest stan serwisowy działający w tle, tzn., że jest on widoczny tylko po podłączeniu się za pomocą kabla do płytki sterującej i uruchomieniu monitora portu szeregowego

- Stan oczekiwania na monety – wrzutnik monet jest aktywny i czeka na wrzucenie przez klienta pieniędzy. Początkowo ekran wyświetla „Wrzuc 5, 2, 1zł”. Po zaakceptowaniu pierwszej monety przez akceptor, pojawia się informacja „Kredyt: x”, gdzie x to wartość wrzuconego przez klienta bilonu. Po osiągnięciu 20zł stan przechodzi w stan wyboru produktu.
- Stan krytyczny - stan, w którym automat jest pusty. Na ekranie wyświetla się informacja „obsługa klienta tel. 511-***-834”
- Stan wyboru produktu – W tym stanie klient dokonuje wyboru produktu spośród dostępnych możliwości A,B,C,D.
- Stan tryb serwisowy został szczegółowo opisany w punkcie 3.4.3 Tryb serwisowy.

3.4. Implementacja

3.4.1. Biblioteki

Do zaprogramowania wyświetlacza LCD z konwerterem magistrali I2C użyłem gotowej biblioteki <Wire.h>. Pozwala ona na komunikację z urządzeniami I2C. Kolejną użytą biblioteką jest <LiquidCrystal_I2C.h>. Służy ona do sterowania wyświetlaczami I2C. Jest również kompatybilna z architekturą AVR, co sprawia, że można jej używać na Arduino Mega 2560. Kolejną użytą biblioteką jest <Stepper.h>, która umożliwia sterowanie jednobiegunowymi lub bipolarnymi silnikami krokowymi.

3.4.2. Główna pętla programu

```

78.      // ===== GŁÓWNA PĘTLA PROGRAMU
79.
80.      void loop(){
81.          trybSerwisowy();
82.
83.          if (cents >= targetcents){
84.              credits = credits + 1;
85.              cents = cents - targetcents;
86.              wybor =1;
87.          }
88.
89.          if(cents==0&&cents<20&&wybor==0){
90.              waitingForCredit();
91.          }
92.
93.          if(cents>0&&cents<20&&wybor==0){
94.              updateCredits();
95.          }
96.
97.          if(wybor==1){
98.              wyborProduktu();
99.              cents = 0;
100.             delay(600);
101.          }
102.
103.             delay(500);
104.             lcd.clear();
105.         }

```

Listing 3.1 Fragment kodu przedstawiający główną pętlę programu

Wchodząc do głównej pętli programu wywołujemy funkcję tryb serwisowy zawartą w stanie włączonym automatu, która w diagramie UML jest opisana jako stan serwisowy i działa w tle. Jeżeli wartość monet w złotówkach równa 0 i mniejsza od 20 oraz zmienna `wybor` nazywana później flagą jest równa 0 wywoływana jest funkcja `waitingForCredit()` zawierająca się w stanie oczekiwania na monety. Po zaakceptowaniu pierwszego bilonu wywoływana jest funkcja `updateCredits()`, która aktualizuje stan wrzuconych kredytów(złotówek). Po osiągnięciu 20 złotych, flaga zostaje ustawiona na 1, a program przechodzi ze stanu oczekiwania na monety do stanu wyboru produktu, jeżeli spełnione jest założenie, że znajduje się jakikolwiek produkt, w przeciwnym wypadku automat znalazłby się w stanie krytycznym. Po naciśnięciu przez klienta przycisku odpowiednio A,B,C lub D obraca się przypisany do guzika silnik, a zmienna `cents` odpowiadająca za nasz kredyt jest zerowana i ekran czyszczony, co implikuje powrót do stanu oczekiwania na monety.

3.4.3. Tryb serwisowy

Podłączenie się do trybu serwisowego wymaga od serwisanta posiadania kabla USB typu B oraz urządzenia z zainstalowanym środowiskiem obsługującym monitor portu szeregowego dla Arduino. Kabel powinien być wpięty jedną stroną do portu USB w urządzeniu (np. laptop) oraz drugą stroną (typ B) do płyty sterującej Arduino, co zapewni połączenie między urządzeniami. Po nawiązaniu połączenia zostaje uruchomiony monitor portu szeregowego.

```
111. void trybSerwisowy() {
112.     if(Serial.available() > 0){
113.         if(potwierdzenie == 0){
114.             haslo = Serial.readStringUntil('\n');
115.             if(haslo.equals("1234")){
116.                 potwierdzenie=1;
117.                 Serial.println("****");
118.                 Serial.println("Haslo wpisane poprawnie!\n");
119.                 Serial.println("Opcje:\n");
120.                 Serial.println("1.Stan konta:\n");
121.                 Serial.println("2.Stan produktow:\n");
122.                 Serial.println("3.Uzupelnienie:\n");
123.             }
124.         }
125.         if(haslo!="1234"&&potwierdzenie>0){
126.             Serial.print("Bledne haslo");
127.         }
128.         if(potwierdzenie==1){
129.             opcja = Serial.readStringUntil('\n');
130.             if(opcja.equals("1")){
131.                 Serial.println("1.Stan konta:\n");
132.                 Serial.println(stan_konta);
133.             }
134.             if(opcja.equals("2")){
135.                 Serial.println("2.Stan produktow:\n");
136.                 Serial.println("Produkt A:\n");
137.                 Serial.println(produkty_a);
138.                 Serial.println("Produkt B:\n");
139.                 Serial.println(produkty_b);
140.                 Serial.println("Produkt C:\n");
141.                 Serial.println(produkty_c);
142.                 Serial.println("Produkt D:\n");
```

```

143.         Serial.println(produkty_d);
144.     }
145.     if(opcja.equals("3")){
146.         Serial.println("Stan produktow zostal
uzupelniony\n");
147.         uzupeelnienie = 1;
148.         if(uzupeelnienie=1){
149.             produkty_a = 7;
150.             produkty_b = 7;
151.             produkty_c = 7;
152.             produkty_d = 7;
153.             delay(200);
154.             uzupeelnienie = 0;
155.         }
156.     }
157. }
158. }
159. }

```

Listing 3.2 Fragment kodu przedstawiający działanie trybu serwisowego

Funkcja `trybSerwisowy()` czeka na pobranie znaków z klawiatury poprzez monitor portu szeregowego. Jeżeli znaki (hasło) zostaną wprowadzone poprawnie (w tym przypadku „1234”) otrzymamy komunikat „Hasło wpisane poprawnie!”, po czym na ekranie pojawią się opcje: 1.stan konta, 2.stan produktów, 3.uzupełnienie. W przypadku wpisania błędnego hasła (innego niż „1234”) otrzymamy komunikat „Błędne hasło”. Po wpisaniu odpowiednio 1,2 lub 3 otrzymamy następujące wiadomości:

1. Stan konta – aktualna wartość monet znajdująca się w automacie,
2. Stan produktów – Ilość produktów znajdujących się aktualnie na półce A,B,C,D,
3. Uzupełnienie – Serwisant używa tej opcji, gdy uzupełni fizycznie automat produktami do maksymalnej pojemności tj. 7 produktów dla półek A,B,C,D.

3.4.4. Wybór produktu

Funkcja `wyborProduktu()` przedstawia zachowanie się programu po wybraniu przez klienta odpowiednio przycisków A,B,C lub D. Metoda `Silnik1()` wywołuje funkcję, która sprawia, że wał silnika nr 1 obraca się o 360 stopni zgodnie z ruchem wskazówek zegara, co implikuje obrót sprężyny i wypadnięcie produktu.

```

215. void wyborProduktu() {
216.     if(produkty_a > 0) {
217.         lcd.setCursor(0,0);
218.         lcd.print("A");
219.     }
220.     if (produkty_b >0) {
221.         lcd.setCursor(7,0);
222.         lcd.print("B");
223.     }
224.     if(produkty_c>0) {
225.         lcd.setCursor(0,1);
226.         lcd.print("C");
227.     }
228.     if(produkty_d>0) {
229.         lcd.setCursor(7,1);
230.         lcd.print("D");

```

```

231.     }
232.     if(produkty_a + produkty_b + produkty_c + produkty_d <= 0){
233.         lcd.setCursor(2,0);
234.         lcd.print("Brak produktow");
235.         delay(3000);
236.         lcd.setCursor(0,0);
237.         lcd.print("Obsluga klienta");
238.         lcd.setCursor(0,1);
239.         lcd.print("tel 511-079-834");
240.     }
241.     if(digitalRead(PRZYCISK_A)==HIGH && produkty_a>0){
242.         lcd.clear();
243.         delay(100);
244.         lcd.print("Wybrano: A");
245.         delay(500);
246.         Produkty_a--;
247.         Silnik1();
248.         wybor = 0;
249.     }
250. }

```

Listing 3.3 Fragment kodu przedstawiający implementację funkcji wyborProduktu()

3.4.5. Programowanie wrzutnika monet

Wrzutnik monet został zaprogramowany przez producenta w sposób umożliwiający osobie programującej ustawienie odpowiednich parametrów według indywidualnych potrzeb. Akceptor obsługuje 6 rodzajów monet, których rodzaje można w prosty sposób zaprogramować przy użyciu przycisków i wyświetlacza. Czujnik zastosowany w urządzeniu identyfikuje monety wykorzystując ich średnicę, grubość oraz czas opadania. Szczegółowe testy oraz implementacja monet użytych w projekcie przebiega w następujący sposób:

Typ monety Parametr			
H	20	20	20
P	1	2	5
F	5	5	5

Tabela 3.1 Przypisanie parametrów do typów monet

Opis parametrów:

E – liczba typów monet, które chcemy użyć (możliwy zakres – 1-6).

H – liczba testowych monet, jakie wrzucimy w celu „nauczenia” wrzutnika rozpoznawania ich jako monety docelowe (możliwy zakres - 1-30).

P – liczba impulsów, jakie powinien generować wrzutnik dla indywidualnego typu monety (zakres 1 - 30) zaczynając od najmniejszego nominału.

F – Dokładność (domyślnie 5)

Parametr E wynosi 3 – wrzutnik przyjmuje 3 różne typy monet.

Do obsługi przerwań służy funkcja `attachInterrupt(int, func, mode)`, gdzie:

`int` – numer przerwania. Arduino Mega 2560 obsługuje dwa przerwania (o numerach 0 i 1),

`func` – nazwa funkcji obsługującej przerwanie

`mode` – ustala kiedy przerwanie ma być wywołane

LOW – kiedy na pinie jest stan niski,

CHANGE – kiedy zmienia się stan,

RISING – kiedy zmienia się stan z LOW na HIGH,

FALLING – kiedy zmienia się stan z HIGH na LOW.

W kodzie przedstawia się to następująco:

```
58.attachInterrupt(digitalPinToInterrupt(coinpin), coinInterrupt, FALLING)
```

Listing 3.4 Fragment kodu przedstawiający funkcję `attachInterrupt()`

```
170. void coinInterrupt() {  
171.     if (wybor==0 && cents<=10) {  
172.         cents++;  
173.         Stan_konta++;  
174.     }  
175. }
```

Listing 3.5 Fragment kodu przedstawiający funkcję `coinInterrupt`

Funkcją obsługującą przerwanie jest tutaj `coinInterrupt()`, w której efektem przerwania jest inkrementacja zmiennej `cents` odpowiadającej za dodawanie ilości impulsów oraz zmiennej `stan_konta` zliczającej pieniądze w automacie. Wywołanie tej funkcji nastąpi, gdy stan na pinie zmieni się z HIGH na LOW. Jak widać na rysunku 3.8 dla monety 2zł Arduino odbiera 2 impulsy. Sytuacja wygląda analogicznie dla 1zł – 1 impuls, 5zł – 5 impulsów.



Rys 3.8 Zdjęcie z oscyloskopu ukazujące zmianę stanu dla monety 2zł

3.4.6. Silniki krokowe

Do zaprogramowania silników krokowych użyłem biblioteki `<Stepper.h>`. Producent informuje nas, że jeden obrót silnika to 32 kroki. Dodatkowo silnik wyposażony jest w przekładnię, która składa się z 64 stopni. Podsumowując aby zauważyć pełny obrót silnika razem z przekładnią należy wykonać: $32 \cdot 64 = 2048$ kroków. Aby Arduino poprawnie odczytywało sygnały trzeba zdefiniować odpowiednio połączenia pinów, co przedstawia poniższy kod.

```

12.     #include <Stepper.h>
13.     #define STEPS_NUMBER 32
14.     Stepper mojSilnik1(STEPS_NUMBER, 22, 23, 24, 25);
15.     Stepper mojSilnik2(STEPS_NUMBER, 26, 27, 28, 29);
16.     Stepper mojSilnik3(STEPS_NUMBER, 30, 31, 32, 33);
17.     Stepper mojSilnik4(STEPS_NUMBER, 34, 35, 36, 37);

```

Listing 3.6 Z Fragment kodu przedstawiający przypisanie pinów w Arduino, którymi sterujemy silnikami krokowymi

Aktywacja silników krokowych odbywa się za pomocą odpowiednio funkcji: `void Silnik1()`, `void Silnik2()`, `void Silnik3()`, `void Silnik4()`. Zawartość tych funkcji różni się jedynie numeracją w nazwie, zatem poniżej przedstawię kod funkcji `void Silnik1()`.

```

290.     void Silnik1() {
291.         mojSilnik1.setSpeed(700);
292.         mojSilnik1.step(2048);
293.         delay(2000);
294.     }

```

Listing 3.7 Fragment kodu przedstawiający implementację funkcji `Silnik1()`,

Linijka `mojSilnik1.setSpeed(700);` oznacza ustawienie prędkości silnika. W tym przypadku odpowiednia liczba to 700. Jest ona wyrażana w jednostce RPM – revolution per minute i oznacza liczbę pełnych obrotów w ciągu minuty wokół osi.

4. Testy i wnioski

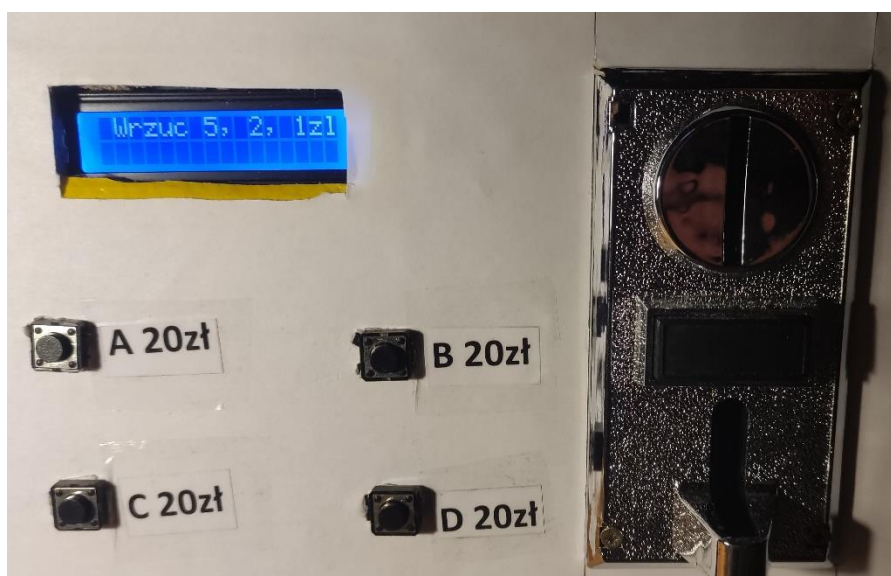
4.1. Testowanie automatu

1. Uruchamiamy automat poprzez podłączenie kabla zasilającego pod napięcie 230V. Ekran jest podświetlony na niebiesko, a kontrolki na płycie Arduino się zapaliły, co oznacza, że płyta jest zasilana.



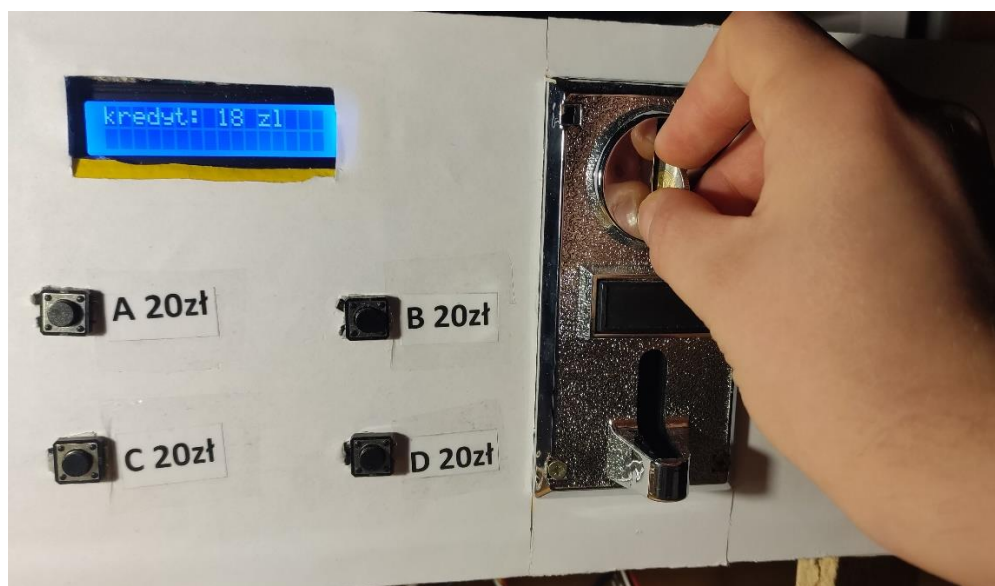
Rys. 4.1 Automat Vendingowy

2. Wrzucamy monety 1,2 lub 5zł. Automat przyjmuje monety.



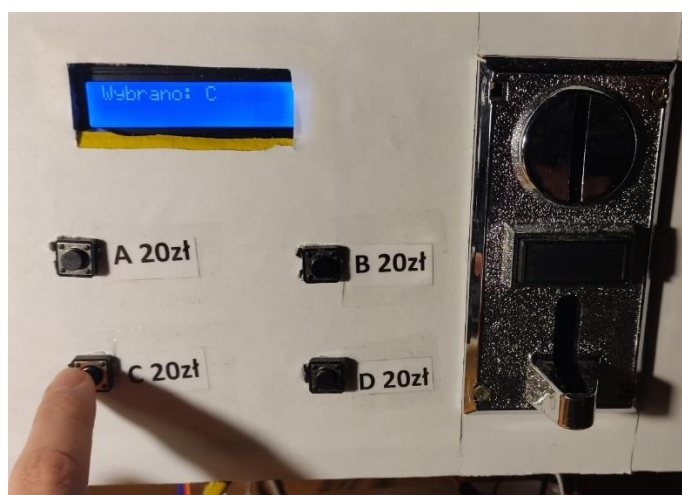
Rys. 4.2 Frontowa ściana automatu

3. Wrzucamy monety, aż osiągniemy wartość produktu (w tym przypadku 20zł).



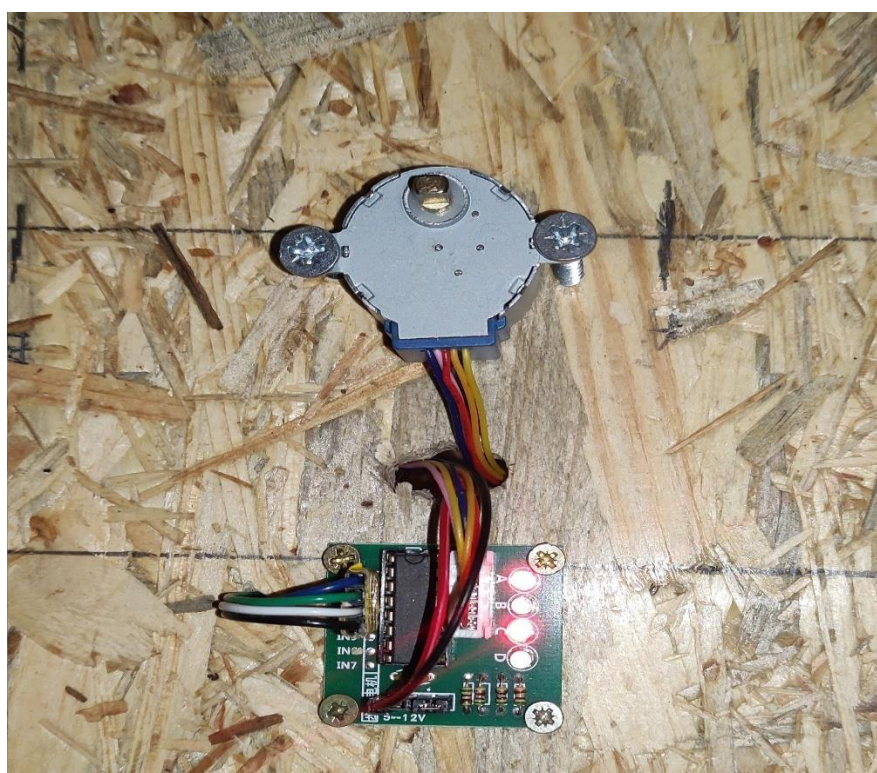
Rys. 4.3 Automat w stanie przyjmowania monet

4. Naciskamy jeden z czterech przycisków (w tym przypadku C).



Rys. 4.4 Ekran po wybraniu przycisku C

5. Na ekranie pojawia się komunikat, że wybrano produkt „C”.



Rys. 4.5 Silnik wykonujący obrót o 360 stopni

6. Diody od sterownika silnika zapalają się, a silnik wykonuje obrót o 360 stopni.

4.2. Wnioski

Głównym problemem poruszonym w pracy było zaprogramowanie komponentów. Część programistyczna została wykonana i spełnia zakładane założenia.

Najbardziej złożony problem stanowiło odpowiednie połączenie wrzutnika monet i zaprogramowanie przerwań. Dowiedziałem się, że aby wrzutnik monet prawidłowo odczytywał impulsy ukazane na rysunku 3.8 trzeba skorzystać z rezystora podciągającego (tzw. Pull up) – w tym przypadku jest to rezystor $10\ 000\ \Omega$, który ma za zadanie niwelować zakłócenia z powietrza.

Niskie koszty komponentów użytych w pracy zmusiły mnie do zrezygnowania z początkowych założeń realizacji mechanicznej, takich jak montaż spiral, czy półek dla odpowiednio silników A,B,C,D.

Mechaniczna realizacja automatu znacząco wykracza poza ramy tego projektu, którego przedmiotem było zaprojektowanie go od strony informatycznej i algorytmicznej

Podsumowując, istnieje wiele możliwości na rozwinięcie projektu tej maszyny vendingowej. W finalnej wersji widać, że arduino MEGA pozostawia wiele niewykorzystanych pinów, co sprawia, że projekt można rozwinąć w wiele różnych kierunków. Jednym z przykładów może być dołączenie modułu wifi bądź modułu bluetooth, co rozwinięłoby model oferowanych płatności o karty kredytowe, czy płatności OFFLINE. Ciekawym rozwiązaniem mógłby być również montaż głośnika, który jest wymagany do realizacji nowoczesnych płatności za pomocą tokena zapisanego w pliku dźwiękowym.

Literatura

- [1] <https://www.strefavendingu.pl/2020/01/15/jak-dawniej-wygladaly-automaty-samoobslugowe-zobacz-pierwszy-automat-vendingowy/>
- [2] http://portalvendingowy.pl/vending/wydawanie_produkow
- [3] <https://allegro.pl/oferta/lapa-szczescia-lapacz-maskotek-toy-story-iii-7642813511>
- [4] <http://modernvending.pl>
- [5] http://portalvendingowy.pl/vending/systemy_platnicze
- [6] <https://scanova.io/blog/qr-codes-on-vending-machine/>
- [7] <https://techcrunch.com/2013/04/14/alipay-launches-sound-wave-mobile-payments-system-in-beijing-subway/>
- [8] <https://neorobot.pl/pl/Arduino-Mega-2560-Rev3.html>
- [9] https://www.amazon.com/BLEE-Different-Selector-Acceptor-Vending/dp/B01M7XZBUR/ref=sr_1_4?s=toys-and-games&ie=UTF8&qid=1545507014&sr=1-4&keywords=coin+acceptor
- [10] <https://botland.com.pl/wyswietlacze-alfanumeryczne-i-graficzne/224-wyswietlacz-lcd-2x16-znakow-niebieski-5903351241380.html>
- [11] <https://botland.com.pl/silniki-krokowe/3480-silnik-krokowy-z-przekladnia-28byj-48-5v01a003nm-ze-sterownikiem-uln2003-openplatform-5903351241458.html>

Data ostatniego wglądu w źródła literatury: 02.02.2022r. godz. 12:00.