



**POLITECHNIKA KRAKOWSKA im. T. Kościuszki**  
Wydział Inżynierii Elektrycznej i Komputerowej  
**Katedra E-1**



Kierunek studiów: Informatyka w Inżynierii Komputerowej

Specjalność: Brak

STUDIA NIESTACJONARNE

## **PRACA DYPLOMOWA** INŻYNIERSKA

**Artur MIGACZ**

PROJEKT MINI MASZYNY VENDINGOWEJ STEROWANEJ ZA  
POMOCĄ ARDUINO

DESIGN OF A MINI VENDING MACHINE CONTROLLED BY  
ARDUINO

Opiekun pracy:  
*Dr inż. Sławomir Bąk*

Kraków, 2025

# SPIS TREŚCI

Streszczenie w języku polskim i angielskim.....	3
Wykaz słów kluczowych w języku polskim i angielskim.....	3
1. WSTĘP.....	4
1.1. Wprowadzenie.....	5
1.2. Cel i zakres pracy.....	6
1.3. Przegląd istniejących rozwiązań.....	8
2. PODSTAWY TEORETYCZNE ZWIĄZANE Z TEMATEM PRACY.....	10
2.1. Automaty vendingowe.....	10
2.2. Systemy mikrokontrolerowe.....	11
2.3. Silniki krokowe.....	12
2.4. Identyfikacja RFID.....	13
2.5. Interfejs I <sup>2</sup> C oraz ekspandery portów.....	14
2.6. Obsługa wejść użytkownika.....	14
2.7. Wyświetlacz LCD.....	15
2.8. Modelowanie 3D oraz druk 3D.....	16
3. OPIS REALIZACJI TEMATU PRACY.....	18
3.1. Przygotowanie Arduino 101.....	18
3.2. Przygotowanie i zainstalowanie komponentów elektronicznych.....	18
3.2.1. Czytnik RFID RC522.....	18
3.2.2. Wyświetlacz LCD 2004.....	19
3.2.3. Przyciski, brzęczyk oraz magistrala I <sup>2</sup> C.....	19
3.2.4. Silniki krokowe.....	21
3.2.5. Optymalizacja przewodów.....	22
3.3. Stworzenie i wykorzystanie grafiki do kart RFID.....	23
3.4. Zaprojektowanie obudowy i wydruk 3D.....	25
3.5. Montaż urządzenia i testy końcowe.....	30
4. NAJWAŻNIEJSZE ELEMENTY OPROGRAMOWANIA.....	30
4.1. Pętla główna i logika transakcji.....	31
4.2. Zarządzanie kredytami użytkownika.....	32
4.3. Zarządzanie produktami.....	32
4.4. Wydawanie produktów.....	33
5. PODSUMOWANIE.....	34
5.1. Osiągnięte cele projektu.....	34
5.2. Możliwości rozwoju i usprawnienia projektu.....	35
5.3. Podsumowanie kosztów projektu.....	36
5.4. Wnioski końcowe.....	36
BIBLIOGRAFIA.....	37
SPIS ILUSTRACJI.....	38

## DODATKI

## **Streszczenie w języku polskim i angielskim**

Projekt polegał na zaprojektowaniu i wykonaniu mini maszyny *vendingowej* sterowanej przez mikrokontroler Arduino, umożliwiającej zakup produktów przy użyciu kart RFID i wewnętrznego systemu kredytowego. Urządzenie łączy elementy mechaniki, elektroniki oraz programowania, tworząc funkcjonalny prototyp o charakterze edukacyjnym i demonstracyjnym.

The project consisted of designing and constructing a mini vending machine controlled by an Arduino microcontroller, enabling product purchases using RFID cards and an internal credit system. The device combines elements of mechanics, electronics, and programming, resulting in a functional prototype with an educational and demonstrative character.

## **Wykaz słów kluczowych w języku polskim i angielskim**

Słowa kluczowe:

maszyna *vendingowa*, mikrokontroler Arduino, RFID, system kredytowy, sterowanie elektroniczne, silniki krokowe, system wbudowany, układ elektroniczny

Keywords:

vending machine, Arduino microcontroller, RFID, credit system, electronic control, step motors, embedded system, electronic circuit

## 1. WSTĘP

Współczesne automaty *vendingowe* stanowią integralny element infrastruktury miejskiej, biurowej oraz komercyjnej. Dzięki swojej dostępności, prostocie obsługi i zdolności do pracy w trybie samoobsługowym, są wykorzystywane do sprzedaży szerokiej gamy produktów – od napojów i przekąsek, przez drobne akcesoria elektroniczne, aż po środki ochrony osobistej. Postępująca automatyzacja oraz rosnące możliwości mikrokontrolerów i otwartych platform sprzętowych, takich jak Arduino, umożliwiają projektowanie oraz realizację funkcjonalnych modeli takich urządzeń również w celach edukacyjnych, hobbyistycznych lub prototypowych.

Niniejsza praca inżynierska ma na celu zaprojektowanie, wykonanie oraz oprogramowanie miniaturowej wersji automatu *vendingowego*, sterowanego za pomocą mikrokontrolera Arduino. Projekt ten opiera się na wykorzystaniu popularnych, ogólnodostępnych komponentów elektronicznych, takich jak: silniki krokowe, wyświetlacz LED, czytnik kart RFID (Radio Frequency Identification), brzęczyk, przyciski oraz ekspandery portów I/O PCF8574. Głównym zadaniem maszyny będzie umożliwienie użytkownikowi wyboru i zakupu jednego z trzech dostępnych produktów przy użyciu dedykowanej kart RFID [1].

W odróżnieniu od komercyjnych urządzeń *vendingowych*, które operują na rzeczywistych płatnościach gotówkowych lub bezgotówkowych, niniejszy projekt wykorzystuje **wewnętrzny system kredytowy**. Każdy użytkownik posiada przypisaną do swojego konta określoną liczbę kredytów, które mogą być wykorzystywane do zakupu produktów. Transakcje odbywają się poprzez automatyczne odjęcie odpowiedniej liczby kredytów z konta użytkownika w momencie dokonania zakupu. System ten upraszcza mechanizmy płatności, jednocześnie umożliwiając wprowadzenie podstawowych funkcji autoryzacji, kontroli dostępu oraz ewidencji użytkowników.

Urządzenie zostało zaprojektowane jako modułowy, edukacyjny prototyp, którego obudowa zostanie wykonana techniką druku 3D przy użyciu programu Autodesk Fusion [2]. Całość oprogramowania zostanie napisana w języku C++ z wykorzystaniem środowiska programistycznego Arduino IDE [3], co zapewnia wysoką elastyczność oraz możliwość łatwej modyfikacji i rozbudowy funkcjonalności. Projekt ten ma również na celu zaprezentowanie praktycznego zastosowania technologii mikrokontrolerowych w kontekście automatyki i interakcji człowiek–maszyna.

## 1.1. Wprowadzenie

Automaty *vendingowe* są przykładem systemów zautomatyzowanych, które zyskały ogromną popularność ze względu na swoją niezawodność, dostępność oraz prostotę użytkowania. Ich działanie opiera się na współpracy wielu podzespołów mechanicznych, elektronicznych oraz informatycznych, co czyni je interesującym obiektem badań i projektów inżynierskich. Wraz z rozwojem technologii mikrokontrolerów oraz dostępnością tanich komponentów, budowa własnych, zminiaturyzowanych urządzeń *vendingowych* stała się możliwa nie tylko dla firm, lecz także dla studentów i pasjonatów elektroniki.

Realizacja niniejszego projektu została podyktowana chęcią stworzenia funkcjonalnego, edukacyjnego prototypu automatu *vendingowego*, który demonstruje podstawowe zasady działania takich urządzeń, przy jednoczesnym ograniczeniu kosztów i uproszczeniu konstrukcji. Główną ideą było opracowanie systemu, który będzie działał na zasadzie wewnętrznego systemu kredytowego, z pominięciem rzeczywistych transakcji pieniężnych, co znaczco upraszcza warstwę sprzętową oraz programistyczną.

W projekcie wykorzystano platformę Arduino, która dzięki swojej otwartości, dostępności bibliotek oraz wsparciu społeczności, idealnie nadaje się do szybkiego prototypowania układów sterowania. W ramach realizacji zaprojektowano również obudowę urządzenia w programie Autodesk Fusion, a następnie wydrukowano ją przy użyciu drukarki 3D, co pozwoliło na dopasowanie wszystkich elementów w sposób estetyczny i funkcjonalny.

Praca została podzielona na kilka głównych etapów:

- Analiza założeń funkcjonalnych i wybór komponentów,
- Zaprojektowanie schematu połączeń elektrycznych,
- Implementacja programu sterującego w środowisku Arduino IDE
- Wykonanie modelu 3D obudowy,
- Przeprowadzenie testów oraz analiza poprawności działania systemu.

Projekt ten, choć niewielki pod względem fizycznym, porusza wiele aspektów istotnych dla nowoczesnej inżynierii – od mechaniki, przez elektronikę, aż po programowanie systemów wbudowanych. Jego uniwersalność sprawia, że może stanowić punkt wyjścia do dalszych badań i rozszerzeń, takich jak komunikacja bezprzewodowa, integracja z bazą danych czy zdalne zarządzanie urządzeniem przez Internet.

## 1.2. Cel i zakres pracy

Celem niniejszej pracy inżynierskiej jest zaprojektowanie, wykonanie i oprogramowanie miniaturowej maszyny *vendingowej*, sterowanej za pomocą mikrokontrolera Arduino, działającej w oparciu o wewnętrzny system kredytowy oraz identyfikację użytkowników za pomocą technologii RFID. Urządzenie ma umożliwiać użytkownikom wybór i zakup jednego z trzech produktów pod warunkiem, że użytkownik posiada wymaganą ilość kredytów.

Dodatkowym celem niniejszej pracy jest zaprojektowanie dedykowanej obudowy przy użyciu programu Autodesk Fusion oraz wykonanie jej przy użyciu technologii druku 3D, co pozwoli na stworzenie fizycznego prototypu w pełni zintegrowanego z elektroniką.

Zakres pracy obejmuje realizację wszystkich etapów związanych z opracowaniem urządzenia – od koncepcji i projektowania, aż po budowę i testowanie prototypu.

W szczególności zakres obejmuje:

- **Analizę wymagań funkcjonalnych urządzenia**, w tym założeń dotyczących liczby produktów, sposobu autoryzacji, sterowania oraz logiki sprzedaży;
- **Wybór i integrację odpowiednich komponentów elektronicznych**, takich jak:
  - Mikrokontroler Arduino 101
  - Silniki krokowe 28BYJ-48 wraz ze sterownikiem ULN2003
  - Wyświetlacz LED
  - Czytnik kart RFID RC522
  - Brzęczyk
  - Przyciski,
  - Eks pandery portów PCF8574 (dla rozszerzenia wejść/wyjść);
- **Opracowanie schematu połączeń elektrycznych** oraz rozmieszczenia komponentów;
- **Implementację programu sterującego w języku C++ w środowisku Arduino IDE**, w tym:
  - Obsługę czytnika RFID,
  - Zarządzanie systemem kredytowym (przechowywanie i aktualizacja salda),
  - Logikę wyboru produktu i jego wydania,
  - Sterowanie silnikami krokowymi,
  - Sygnalizację dźwiękową i wizualną;

- **Zaprojektowanie modelu 3D obudowy w programie Autodesk Fusion,** dostosowanej do rozmiarów poszczególnych podzespołów;
- **Wydruk elementów obudowy** przy użyciu drukarki 3D;
- **Testowanie urządzenia** pod względem funkcjonalnym, logicznym i mechanicznym;
- **Ocenę wyników oraz wskazanie możliwych kierunków dalszego rozwoju projektu.**

Koncepcyjna grafika mini maszyny *vendingowej* została zaprezentowana na Rys. 1.1



Rys. 1.1 Grafika koncepcyjna

### **1.3. Przegląd istniejących rozwiązań**

Współczesne automaty *vendingowe* są szeroko rozpowszechnione i stale rozwijane zarówno pod względem technologicznym, jak i funkcjonalnym. Większość dostępnych na rynku komercyjnych rozwiązań opiera się na zaawansowanych systemach płatności bezgotówkowych, zdalnym monitorowaniu stanów magazynowych oraz rozbudowanych interfejsach użytkownika. W niniejszym podrozdziale przedstawiono krótki przegląd istniejących rozwiązań dostępnych komercyjnie oraz projektów hobbyistycznych i edukacyjnych, które stały się inspiracją dla opracowania własnego prototypu mini maszyny *vendingowej*.

#### **Komercyjne automaty *vendingowe***

Nowoczesne automaty *vendingowe*, stosowane na szeroką skalę w przestrzeni publicznej, centrach handlowych i biurach, to rozbudowane urządzenia zawierające liczne systemy wspomagające pracę. Najczęściej wykorzystują one:

- terminale płatności zbliżeniowej (NFC, karty płatnicze, telefony),
- zaawansowane systemy telemetryczne (np. GSM, Wi-Fi) do monitorowania poziomu zapasów i usterek,
- dotykowe interfejsy graficzne,
- mechanizmy wydawania produktów dostosowane do wielu typów opakowań i rozmiarów.

Urządzenia te są najczęściej projektowane z myślą o wysokiej niezawodności, dużej pojemności oraz integracji z systemami zarządzania sprzedażą. Ze względu na ich rozbudowaną konstrukcję oraz wysokie koszty komponentów, wdrożenie tego typu urządzeń wiąże się z istotnymi nakładami inwestycyjnymi oraz koniecznością zapewnienia odpowiedniego zaplecza serwisowego i technicznego.

#### **Projekty edukacyjne i hobbyistyczne**

W środowiskach akademickich oraz wśród entuzjastów elektroniki i programowania, pojawiła się wiele realizacji uproszczonych maszyn *vendingowych*, zbudowanych w oparciu o platformy mikrokontrolerowe, takie jak Arduino, Raspberry Pi czy ESP32 [1]. Projekty te mają zwykle na celu:

- demonstrację podstawowych zasad automatyki i sterowania,
- naukę programowania systemów wbudowanych,
- zapoznanie się z technikami druku 3D i modelowania mechanicznego,
- rozwój umiejętności integracji sprzętu i oprogramowania.

## **Wnioski z przeglądu**

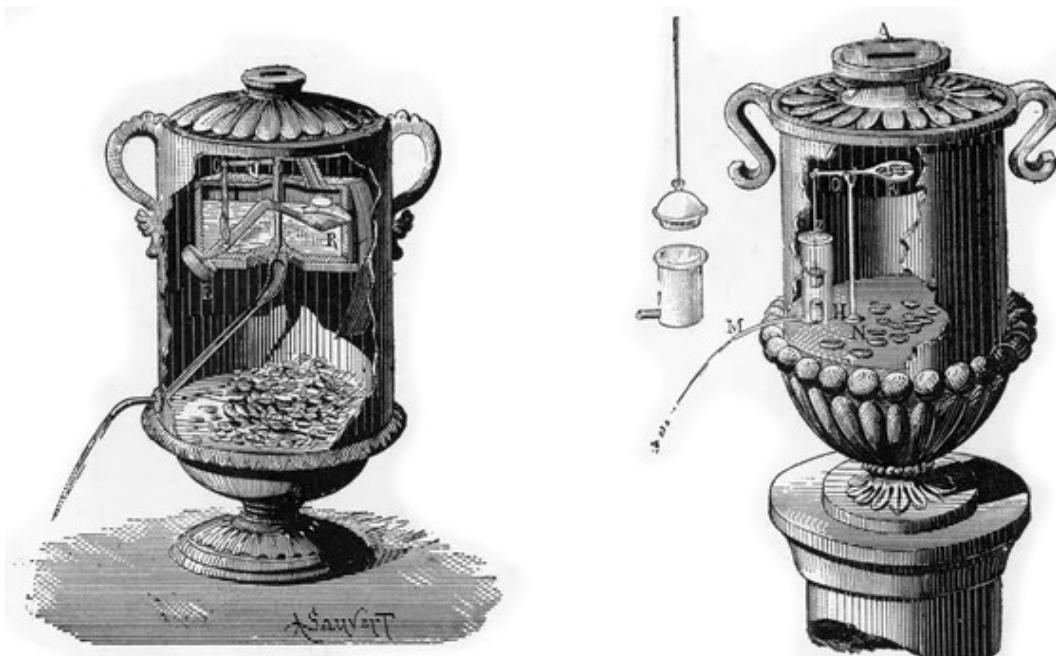
Na podstawie przeglądu istniejących rozwiązań można zauważyc istotne różnice pomiędzy podejściem komercyjnym a edukacyjnym. Komercyjne maszyny vendingowe oferują szeroki zakres funkcji, lecz są zbyt kosztowne i skomplikowane w kontekście naukowym i dydaktycznym. Z kolei projekty edukacyjne i amatorskie, choć uproszczone, pozwalają na efektywną naukę zasad działania zautomatyzowanych systemów. Zaprojektowany w ramach niniejszej pracy prototyp czerpie inspirację właśnie z tej drugiej grupy – stanowiąc kompromis pomiędzy funkcjonalnością, a prostotą wykonania i niskimi kosztami wdrożenia.

## 2. PODSTAWY TEORETYCZNE ZWIĄZANE Z TEMATEM PRACY

W tym rozdziale przedstawiono najważniejsze zagadnienia teoretyczne niezbędne do realizacji projektu mini maszyny *vendingowej*, w tym podstawy działania mikrokontrolerów [1], sterowania silnikami krokowymi, technologii RFID, komunikacji poprzez magistralę I<sup>2</sup>C, obsługi urządzeń peryferyjnych oraz modelowania i druku 3D [2].

### 2.1. Automaty vendingowe

Automaty *vendingowe* (ang. *vending machines*) to samoobsługowe urządzenia umożliwiające sprzedaż różnorodnych produktów bez udziału personelu. Ich głównym celem jest szybka i wygodna dystrybucja towarów – najczęściej napojów czy przekąsek. Pierwsze maszyny vendingowe pojawiły się już w starożytnym Egipcie, gdzie służyły do wydawania świętej wody po wrzuceniu monety. Grafika przedstawiająca działanie takiej maszyny została ukazana na Rys. 2.1



Rys. 2.1 Starożytna maszyna wydająca świętą wodę

Współczesne automaty *vendingowe* są znacznie bardziej zaawansowane technologicznie niż ich pierwowzory – coraz częściej wykorzystują połączenia bezprzewodowe (Wi-Fi, GSM), zdalne zarządzanie, a także możliwość personalizacji oferty.

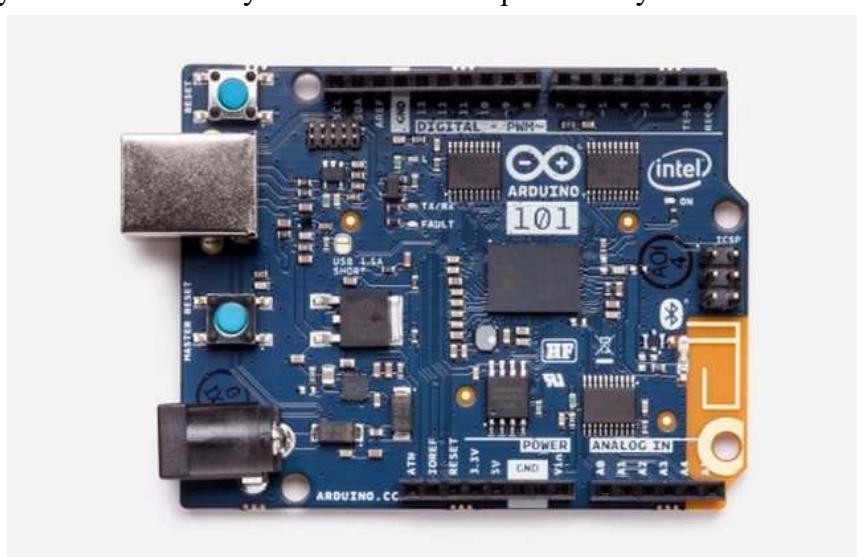
## 2.2. Systemy mikrokontrolerowe

Mikrokontroler to niewielki, zintegrowany układ scalony zawierający w swojej strukturze jednostkę centralną (CPU), pamięć operacyjną (RAM), pamięć programu (ROM/Flash), a także zestaw peryferiów takich jak liczniki, porty wejścia/wyjścia, interfejsy komunikacyjne (np. I<sup>2</sup>C, SPI, UART [3]) oraz przetworniki analogowo-cyfrowe (ADC [2]). Dzięki tej uniwersalnej strukturze mikrokontrolery są podstawowym elementem tzw. systemów wbudowanych (embedded systems), stosowanych powszechnie w automatyce, robotyce, urządzeniach gospodarstwa domowego, pojazdach, a także w sprzęcie medycznym czy konsumenckim.

Zaletą mikrokontrolerów jest możliwość pełnienia funkcji sterownika urządzeń elektronicznych w sposób niezależny, bez potrzeby stosowania zewnętrznego komputera. Programowanie mikrokontrolerów odbywa się zwykle w językach niskiego poziomu, takich jak C lub C++, co pozwala na dokładną kontrolę nad działaniem systemu i zasobami sprzętowymi.

Arduino to otwarta platforma prototypowania elektronicznego, której celem jest ułatwienie tworzenia projektów opartych na mikrokontrolerach, zarówno przez profesjonalistów, jak i hobbystów czy studentów. Podstawowymi elementami systemu Arduino są: płytka z mikrokontrolerem, środowisko programistyczne Arduino IDE oraz bogaty ekosystem bibliotek i dokumentacji. Główną zaletą Arduino jest uproszczony interfejs programowania oraz możliwość szybkiego prototypowania urządzeń elektronicznych.

Arduino 101 (Rys. 2.2), znane również jako Genuino 101, to jedna z bardziej zaawansowanych płyt w rodzinie Arduino. Została zaprojektowana we współpracy z firmą Intel i oparta na mikrokontrolerze Intel® Curie™, który łączy w sobie 32-bitowy rdzeń x86 oraz współdzielony mikrokontroler ARC.



Rys. 2.2 Płytki Arduino 101

## 2.3. Silniki krokowe

Silniki krokowe to rodzaj silników elektrycznych, które wykonują ruch obrotowy w sposób skokowy – obracającą się o ścisłe określony kąt (tzw. krok) w odpowiedzi na impuls sterujący. Dzięki temu możliwe jest bardzo precyzyjne kontrolowanie pozycji wału silnika bez konieczności stosowania czujników położenia. Silniki krokowe są powszechnie wykorzystywane w automatyce, drukarkach 3D, robotyce, sterowaniu zaworami oraz maszynach *vendingowych*, gdzie wymagane jest powtarzalne i kontrolowane przemieszczanie elementów mechanicznych.

Silniki te dzielą się głównie na dwa typy:

- **Unipolarne** – prostsze w sterowaniu, wymagają mniejszej liczby tranzystorów sterujących, ale mają mniejszy moment obrotowy [1][4].
- **Bipolarne** – bardziej efektywne pod względem momentu obrotowego, ale wymagają bardziej zaawansowanego układu sterującego [1][4].

Silnik **28BYJ-48** (Rys. 2.3) który został wykorzystany w projekcie, to popularny, niskonapięciowy silnik krokowy unipolarny, szeroko stosowany w projektach edukacyjnych i prototypowych. Charakteryzuje się prostotą obsługi, niskim poborem mocy oraz wysoką dostępnością.

**Główne parametry techniczne:**

- Napięcie zasilania: **5 V DC**
- Typ: **unipolarny, 4-fazowy**
- Liczba kroków: **2048 kroków na pełen obrót**
- Prąd fazy: **~240 mA**

Dzięki wbudowanej przekładni, silnik ten oferuje wysoki moment obrotowy przy stosunkowo niskiej prędkości, co czyni go idealnym do zastosowań wymagających dużej siły i precyzji, takich jak mechanizmy wydawania produktów.



Rys. 2.3 Silnik krokowy 28BYJ-48

## 2.4. Identyfikacja RFID

RFID (ang. Radio Frequency Identification) [1][5] to technologia bezprzewodowej identyfikacji obiektów, wykorzystująca fale radiowe do przesyłania danych pomiędzy transponderem (tzw. tagiem) a czytnikiem. Każdy tag RFID zawiera unikalny identyfikator, który może być odczytany przez czytnik z określonej odległości, bez potrzeby fizycznego kontaktu.

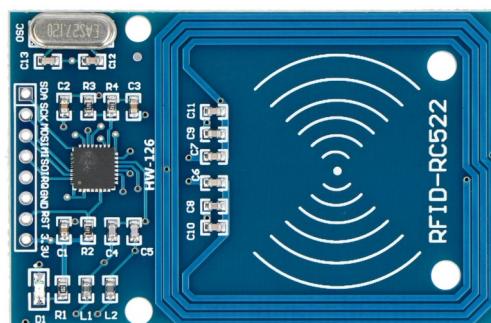
Systemy RFID dzielą się na trzy główne elementy:

- **Tag (transponder)** – mała karta, brelok lub chip z wbudowaną anteną; może być pasywny (bez zasilania) lub aktywny (z własnym źródłem energii).
- **Czytnik RFID** – urządzenie nadawczo-odbiorcze, które emituje sygnał i odbiera dane z tagów.
- **System przetwarzający dane** – mikrokontroler lub komputer, który interpretuje dane identyfikacyjne i wykonuje odpowiednie operacje (np. autoryzację dostępu lub naliczenie punktów).

Moduł **RC522** [5] (Rys. 2.4) który został wykorzystany w projekcie, to popularny czytnik RFID pracujący na częstotliwości **13,56 MHz**, zgodny ze standardem **ISO/IEC 14443 A**, obsługującym m.in. karty typu **MIFARE**. Jest szeroko stosowany w projektach opartych na Arduino ze względu na niski koszt, niewielkie rozmiary oraz łatwą integrację programistyczną.

**Podstawowe cechy RC522:**

- Zasilanie: **3,3 V**
- Zasięg odczytu: do **5 cm** (zależnie od typu tagu)
- Wbudowana antena PCB
- Obsługa operacji odczytu i zapisu danych w pamięci tagów



Rys. 2.4 Moduł RFID RC522

## 2.5. Interfejs I<sup>2</sup>C oraz ekspandery portów

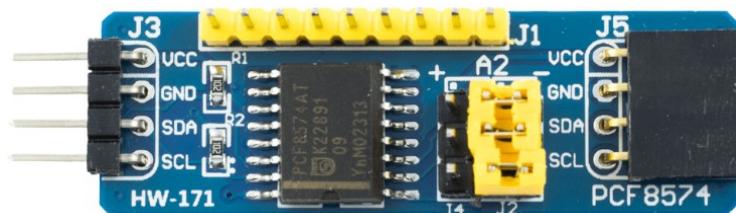
I<sup>2</sup>C (ang. Inter-Integrated Circuit) [1] to szeregowy, synchroniczny interfejs komunikacyjny opracowany przez firmę Philips (obecnie NXP), zaprojektowany do komunikacji między układami scalonymi na niewielkie odległości, najczęściej wewnętrz jednego urządzenia. Umożliwia podłączenie wielu urządzeń peryferyjnych (czujników, ekspanderów, pamięci itp.) do jednego mikrokontrolera przy użyciu tylko dwóch przewodów:

- **SDA (Serial Data Line)** – linia danych,
- **SCL (Serial Clock Line)** – linia zegarowa.

W komunikacji I<sup>2</sup>C urządzenia dzielą się na master (zazwyczaj mikrokontroler) i slave (urządzenia peryferyjne). Każde urządzenie slave ma przypisany unikalny adres, dzięki czemu master może komunikować się z nimi niezależnie.

Zaletą magistrali I<sup>2</sup>C jest prostota okablowania oraz możliwość obsługi wielu urządzeń jednocześnie (nawet kilkudziesięciu), przy użyciu tylko dwóch wspólnych linii sygnałowych.

**PCF8574** [6] (Rys. 2.5) to układ scalony firmy NXP, pełniący rolę **ekspandera portów wejścia/wyjścia** sterowanego przez magistralę I<sup>2</sup>C. Dzięki niemu możliwe jest rozszerzenie liczby dostępnych pinów cyfrowych mikrokontrolera bez konieczności zmiany sprzętu bazowego.



Rys. 2.5 Magistrala I<sup>2</sup>C PCF8574

## 2.6. Obsługa wejść użytkownika

Wejścia cyfrowe w mikrokontrolerach [1][7] służą do odbierania sygnałów binarnych – tzn. mogą przyjmować tylko dwa stany logiczne: **wysoki (HIGH)** lub **niski (LOW)**. Typowym przykładem wejścia cyfrowego jest przycisk, który po naciśnięciu zmienia stan logiczny danego pinu mikrokontrolera. Mikrokontroler może wykryć tę zmianę i zareagować w programie, wywołując odpowiednią funkcję lub wykonując określoną akcję.

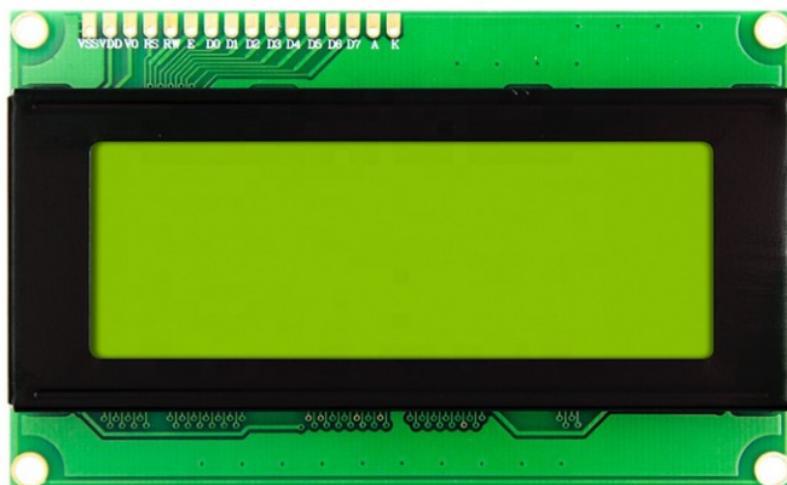
## 2.7. Wyświetlacz LCD

**LCD (ang. Liquid Crystal Display)** [1] to technologia wyświetlania obrazu oparta na właściwościach optycznych ciekłych kryształów. Wyświetlacze LCD są szeroko stosowane w elektronice użytkowej, urządzeniach przemysłowych oraz w systemach wbudowanych, takich jak urządzenia oparte na mikrokontrolerach.

**LCD 2004** [8], który został wykorzystany w tym projekcie to alfanumeryczny wyświetlacz ciekłokrystaliczny, umożliwiający wyświetlanie tekstu w układzie **20 kolumn na 4 wiersze**, wyświetlacz przedstawiony na Rys. 2.6 to jeden z kilku wariantów tego wyświetlacza. Wykorzystuje standardowy sterownik **HD44780**, co zapewnia szeroką kompatybilność z mikrokontrolerami, w tym z platformą Arduino. Wyświetlacz ten ma wbudowany konwenter I<sup>2</sup>C, który ułatwia korzystanie z wyświetlacza za pomocą protokołu komunikacyjnego I<sup>2</sup>C.

### Podstawowe parametry LCD 2004:

- Format: **20x4** (20 znaków w 4 wierszach)
- Napięcie zasilania: **5 V DC**
- Podświetlenie: **LED**
- Wymiary: ok. **98 mm x 60 mm**
- Sterownik: **HD44780**
- Interfejs: **I<sup>2</sup>C** z modułem PCF8574



Rys. 2.6 Wyświetlacz LCD 4x20

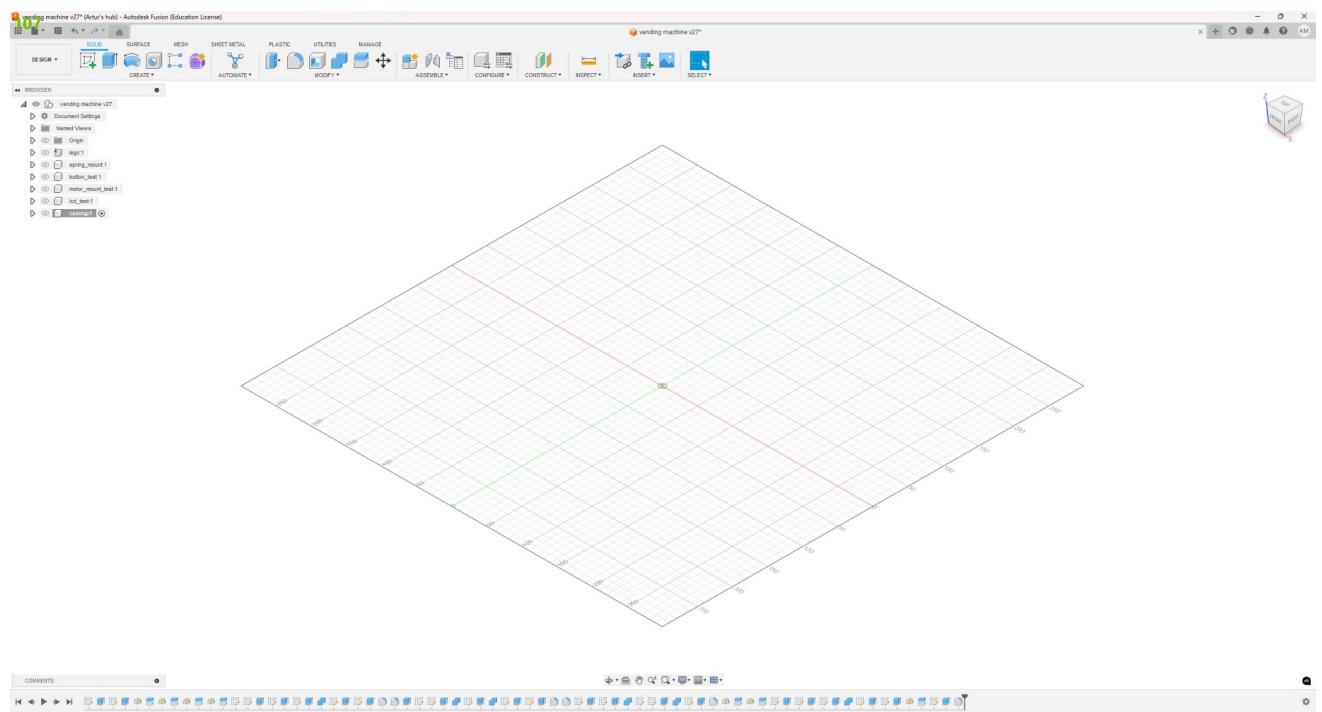
## 2.8. Modelowanie 3D oraz druk 3D

**Modelowanie 3D** [9] polega na tworzeniu cyfrowych modeli obiektów trójwymiarowych, które odwzorowują ich rzeczywiste kształty, wymiary i funkcje. Proces ten odbywa się w specjalistycznym oprogramowaniu CAD (Computer-Aided Design), które pozwala użytkownikowi projektować i modyfikować obiekty inżynierskie w przestrzeni wirtualnej.

**Programy CAD** takie jak wykorzystany w tym projekcie Autodesk Fusion, który został ukazany na Rys. 2.7, oferują narzędzia do tworzenia:

- Szkiców 2D i przekształcania ich w modele 3D (ekstruzja, obrót, wycięcia)
- Dokładnych wymiarów i tolerancji elementów
- Złożień mechanicznych (ang. *assemblies*)
- Wizualizacji i symulacji działania konstrukcji przed jej fizycznym wykonaniem

**Druk 3D** [2] to proces wytwarzania fizycznych obiektów trójwymiarowych na podstawie modelu cyfrowego, poprzez nakładanie kolejnych warstw materiału, najczęściej tworzywa sztucznego.



Rys. 2.7 Okno projektu w programie Autodesk Fusion

## **Główne etapy druku 3D:**

- **Przygotowanie modelu 3D**

Model zaprojektowany w programie CAD eksportowany jest do formatu **.STL** lub **.3MF**, który reprezentuje powierzchnię bryły za pomocą siatki trójkątów.

- **Slicing – przygotowanie do druku**

Plik STL wczytywany jest do programu typu **slicer** (np. wykorzystany w pracy Cura), gdzie:

- Określa się parametry druku (wysokość warstwy, wypełnienie, temperatury, podpory),
- Generowany jest plik **G-code**, zawierający instrukcje ruchu dla drukarki 3D.

- **Proces drukowania**

Drukarka 3D odczytuje G-code i wytwarza obiekt poprzez topienie i nanoszenie cienkich warstw filamentu (najczęściej PLA) zgodnie z zaplanowaną ścieżką. Warstwy nakładane są kolejno od dołu ku górze.

- **Obróbka końcowa**

Gotowy wydruk może wymagać:

- Usunięcia podpór
- Wygładzenia powierzchni
- Dopasowania elementów ruchomych
- Ewentualnego klejenia lub montażu z innymi częściami

### **3. OPIS REALIZACJI TEMATU PRACY**

Rozdział ten przedstawia szczegółowy przebieg realizacji projektu od strony praktycznej. Zawarte tu informacje obejmują proces montażu komponentów elektronicznych, integracji poszczególnych modułów, projektowania elementów mechanicznych, przygotowania materiałów graficznych oraz testowania gotowego urządzenia. Opis ukazuje kolejne kroki podjęte w trakcie tworzenia funkcjonalnego prototypu wraz z napotkanymi trudnościami i sposobami ich rozwiązania.

#### **3.1. Przygotowanie Arduino 101**

Prace nad projektem rozpoczęto od przygotowania środowiska programistycznego. Na komputerze zainstalowano Arduino IDE [10], umożliwiające pisanie i wgrywanie kodu do mikrokontrolera. Następnie, za pomocą przewodu USB, nawiązano połączenie z płytą Arduino, co pozwoliło na przetestowanie podstawowej komunikacji i rozpoczęcie dalszej konfiguracji podzespołów elektronicznych.

#### **3.2. Przygotowanie i zainstalowanie komponentów elektronicznych**

W tej części opisano proces podłączania i konfiguracji głównych komponentów elektronicznych użytych w projekcie. Omówiono również wyzwania napotkane podczas montażu, metody ich rozwiązania oraz działania podjęte w celu zapewnienia stabilności połączeń i poprawności komunikacji między modułami.

##### **3.2.1. Czytnik RFID RC522**

W pierwszej kolejności podłączono czytnik RFID RC522 do płytki Arduino 101 zrealizowano za pomocą interfejsu SPI. Moduł RFID wymaga podłączenia pięciu głównych linii: zasilania (3.3V i GND) oraz sygnałowych: SCK (zegar), MOSI (dane do modułu), MISO (dane z modułu) i SS (wybór urządzenia). Zgodnie z dokumentacją techniczną, linie te przypisano do odpowiednich pinów cyfrowych Arduino 101, uwzględniając jego specyfikę sprzętową.

Po dokonaniu fizycznego podłączenia, przeprowadzono testy komunikacji, wykorzystując bibliotekę *MFRC522*, która umożliwia obsługę czytnika RFID. Zweryfikowano poprawność działania poprzez odczytanie identyfikatorów kart zbliżanych do modułu.

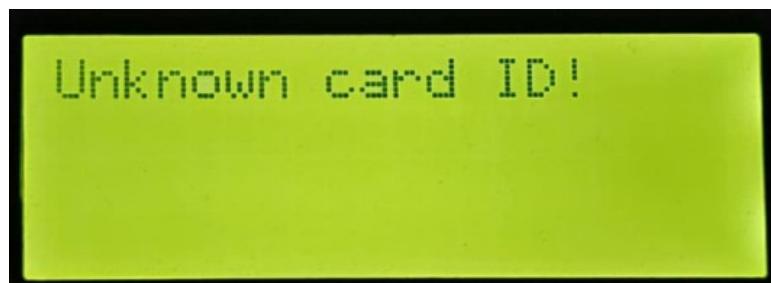
W dalszym etapie prac opracowano klasę *User*, której zadaniem jest przechowywanie informacji o użytkownikach oraz zarządzanie przypisanymi im środkami w wewnętrznym systemie kredytowym. Równolegle stworzono klasę *Item*, służącą do reprezentowania przedmiotów dostępnych w automacie, wraz z ich podstawowymi właściwościami, takimi jak nazwa czy cena. Obie klasy stanowią podstawę logicznej struktury systemu, umożliwiając sprawne zarządzanie transakcjami oraz asortymentem urządzenia.

### **3.2.2. Wyświetlacz LCD 2004**

Następnie przygotowano wyświetlacz LCD 2004 wyposażony w konwerter I<sup>2</sup>C, co umożliwia uproszczoną komunikację z mikrokontrolerem. Wyświetlacz podłączono do płytki Arduino 101 za pomocą czterech przewodów: dwóch zasilających (VCC i GND [1]) oraz dwóch komunikacyjnych (SDA i SCL [1]). Przewody sygnałowe połączono z odpowiednimi pinami magistrali I<sup>2</sup>C na płytce Arduino.

Po wykonaniu połączeń zainstalowano w środowisku Arduino IDE odpowiednią bibliotekę *LiquidCrystal\_I2C*, a następnie przeprowadzono test działania wyświetlacza za pomocą prostego programu inicjalizującego ekran i wyświetlającego przykładowy tekst. Potwierdzono prawidłowe działanie układu oraz poprawność konfiguracji adresu I<sup>2</sup>C.

W kolejnym etapie opracowano klasę *DisplayHandler*, której zadaniem jest usprawnienie zarządzania wyświetlaczem. Klasa ta odpowiada za prezentację komunikatów oraz kontrolę czasu ich wyświetlania, co pozwala na czytelne i uporządkowane przekazywanie informacji użytkownikowi. Przykładem zastosowania może być komunikat o nieznanym identyfikatorze karty RFID, widoczny na Rys 3.1.



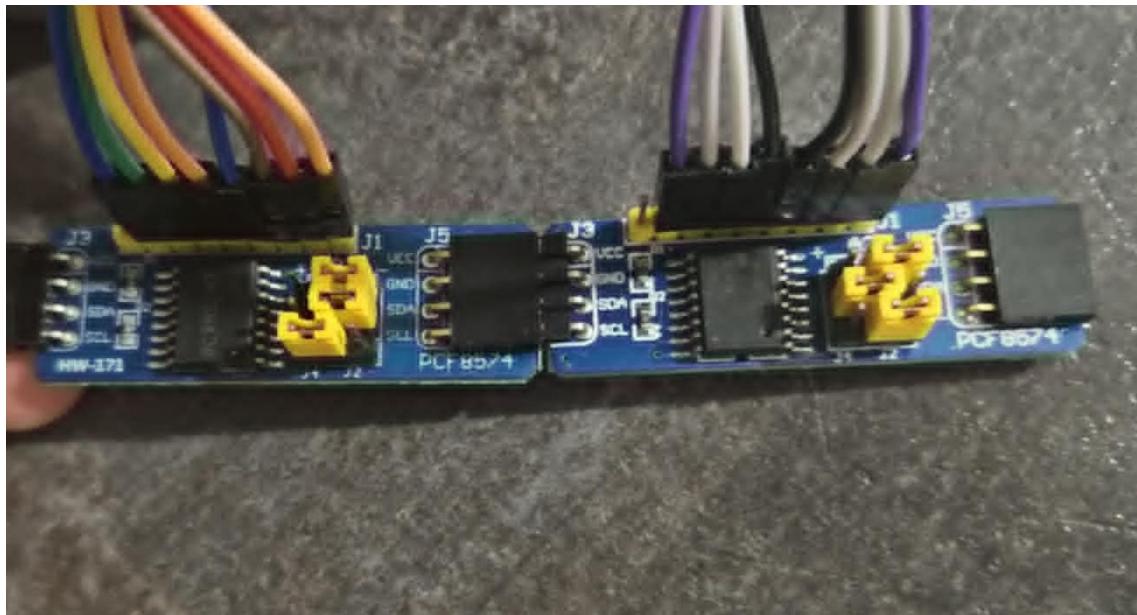
Rys. 3.1 Komunikat oznajmiający o nieznanym identyfikatorze karty

### **3.2.3. Przyciski, brzęczyk oraz magistrala I<sup>2</sup>C**

Podczas podłączania kolejnych komponentów, takich jak przyciski sterujące, brzęczyk oraz dodatkowe moduły komunikacyjne, napotkano ograniczenie wynikające z niewielkiej liczby dostępnych pinów cyfrowych w mikrokontrolerze Arduino 101. Aby umożliwić jednoczesne działanie wszystkich potrzebnych elementów bez konieczności przełączania pinów, zdecydowano się na wykorzystanie magistrali I<sup>2</sup>C (Inter-Integrated Circuit) [1].

Interfejs I<sup>2</sup>C umożliwia komunikację wielu urządzeń za pomocą zaledwie dwóch przewodów – SDA (danych) i SCL (zegara) – co znaczowo ogranicza liczbę wymaganych połączeń na płytce mikrokontrolera. W niniejszym projekcie magistrala ta została wykorzystana do efektywnej obsługi elementów interfejsu użytkownika, takich jak przyciski oraz brzęczyk, a także do sterowania silnikami krokowymi, co pozwoliło zminimalizować zużycie dostępnych pinów wejścia/wyjścia.

Cały układ został zaprojektowany z myślą o możliwie efektywnym wykorzystaniu zasobów mikrokontrolera, przy jednoczesnym zachowaniu pełnej funkcjonalności oraz możliwości dalszej rozbudowy systemu. Adresy urządzeń PCF8574 na magistrali I<sup>2</sup>C zostały fizycznie skonfigurowane na 0x21 oraz 0x22 za pomocą zworek adresowych, co przedstawiono na Rys. 3.2.



Rys. 3.2 Dwie magistrale PCF8574 z różną konfiguracją zworek

Do obsługi przycisków opracowano klasę *ButtonHandler*, której zadaniem jest monitorowanie stanu sygnałów na przypisanych pinach oraz zwracanie identyfikatora wciśniętego przycisku w momencie wykrycia aktywacji.

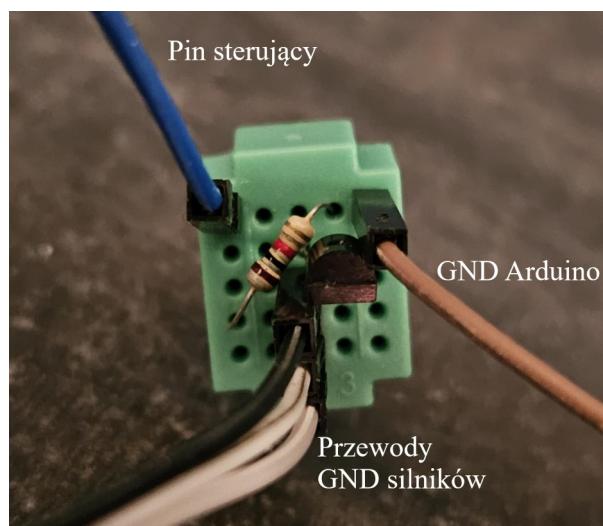
Brzęczyk zastosowany w projekcie został obsłużony za pomocą dedykowanej klasy *BuzzerHandler*, umożliwiającej generowanie sygnałów dźwiękowych w różnych konfiguracjach, dostosowanych do aktualnych potrzeb systemu.

### 3.2.4. Silniki krokowe

Proces montażu silników krokowych rozpoczęto od instalacji i uruchomienia jednego egzemplarza w celu sprawdzenia poprawności jego działania. Pozwoliło to na przetestowanie połączeń oraz weryfikację działania autorskiego oprogramowania sterującego. Do obsługi silników opracowano własną klasę *MotorHandler*, która nie bazuje na dostępnych bibliotekach do sterowania silnikami krokowymi. Zamiast tego cały mechanizm działania został zaimplementowany od podstaw. Sterowanie polega na cyklicznym przesyłaniu sygnału wysokiego do kolejnych cewek silnika, z zachowaniem odstępów czasowych wynoszących 2 milisekundy, co zapewnia płynny i kontrolowany ruch obrotowy.

Po ukończeniu klasy i pomyślnych testach jednego silnika, przystąpiono do montażu pozostałych jednostek. W tym momencie pojawił się problem wynikający z faktu, że silniki były podłączone do magistrali I<sup>2</sup>C, która domyślnie ustawia piny w stan wysoki. Powodowało to przepływ prądu przez wszystkie cewki trzech silników jednocześnie, co skutkowało nadmiernym obciążeniem mikrokontrolera Arduino i uniemożliwiało jego poprawne uruchomienie.

W celu rozwiązania tego problemu zastosowano tranzystor, którego zadaniem było podłączenie silników do masy Arduino dopiero po pełnym uruchomieniu mikrokontrolera. Dzięki temu możliwe było bezpieczne i kontrolowane włączenie zasilania silników dopiero w momencie, gdy piny magistrali I<sup>2</sup>C zostały ustawione w stan niski, co wyeliminowało ryzyko przeciążenia układu. Dodatkowo, pomiędzy tranzystorem a uziemieniem Arduino umieszczono rezystor o wartości 10 kΩ [1], który stabilizuje obwód i zapobiega przypadkowemu przewodzeniu sygnału w stanie nieustalonym. Rys. 3.3 przedstawia fotografię rzeczywistego układu, obrazującą sposób poprowadzenia przewodów do tranzystora.



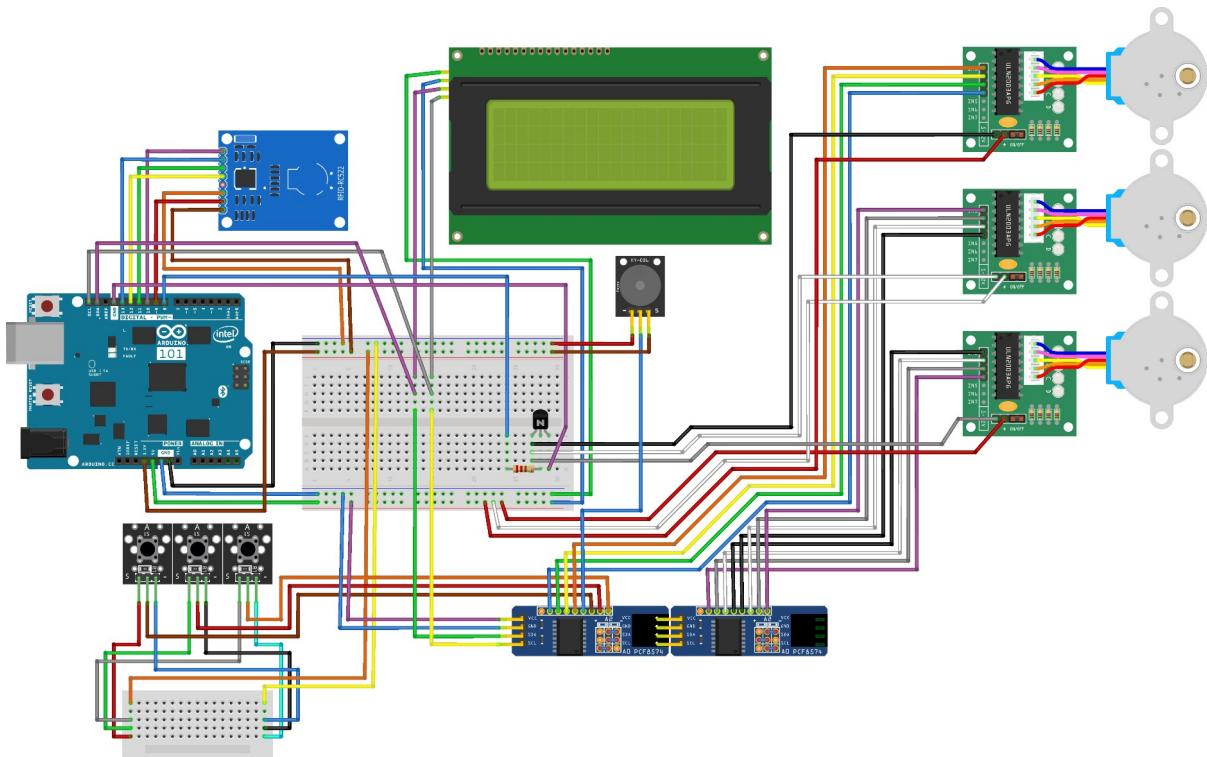
Rys. 3.3 Przedstawienie działania tranzystora w projekcie

### 3.2.5. Optymalizacja przewodów

Po zakończeniu montażu wszystkich komponentów elektronicznych, przystąpiono do optymalizacji połączeń przewodów. Celem tego etapu było zwiększenie przejrzystości układu oraz ułatwienie dalszego montażu całego systemu wewnątrz obudowy. Początkowe połączenia, wykonane głównie w celach testowych, były prowadzone w sposób tymczasowy i niezorganizowany, co mogło utrudniać serwisowanie oraz ograniczać dostęp do poszczególnych elementów.

W ramach optymalizacji uporządkowano przebieg przewodów, skracając je tam, gdzie było to możliwe, oraz grupując zgodnie z funkcją lub lokalizacją w układzie. Do organizacji połączeń wykorzystano również *mini breadboardy* [1], które umożliwiły lepsze rozmieszczenie przewodów i zredukowanie ilości luźnych połączeń. Dodatkowo zastosowano opaski zaciskowe i oznaczenia przewodów, co ułatwia identyfikację poszczególnych linii sygnałowych i zasilających.

Dla pełniejszego zobrazowania struktury całego układu oraz ułatwienia ewentualnych modyfikacji w przyszłości, przygotowano graficzny schemat połączeń wszystkich elementów. Schemat ten został przedstawiony na Rys. 3.4 i prezentuje logiczne powiązania między komponentami systemu w sposób klarowny i uporządkowany.



Rys. 3.4 Graficzna wizualizacja połączeń w projekcie

### 3.3. Stworzenie i wykorzystanie grafiki do kart RFID

W celu nadania całości projektu spójnego i profesjonalnego wyglądu, dla każdego użytkownika przygotowano indywidualną grafikę przypisaną do jego karty RFID. Grafiki te miały na celu, imitowanie rzeczywistych kart identyfikacyjnych pracowników firmy, korzystających z mini maszyny *vendingowej*. Projektowanie odbyło się przy użyciu programu graficznego GIMP, który umożliwił wygodne rozmieszczenie elementów wizualnych, takich jak imię i nazwisko użytkownika, logo Politechniki Krakowskiej oraz twarz wygenerowana przy pomocy sztucznej inteligencji.

Przykładowy wygląd takiej grafiki został zaprezentowany na Rys. 3.5. Do stworzenia realistycznych portretów użytkowników wykorzystano sztuczną inteligencję dostępną na stronie [thispersondoesnotexist.com](http://thispersondoesnotexist.com), która generuje wizerunki osób nieistniejących, lecz wyglądających jak prawdziwe. Dzięki temu możliwe było nadanie każdej karcie unikalnego charakteru bez potrzeby używania rzeczywistych zdjęć.



Rys. 3.5 Przedstawienie projektu graficznego karty

Gotowe projekty zostały wydrukowane na papierze samoprzylepnym, który umożliwił łatwe i estetyczne przymocowanie grafik do kart RFID. Po wydrukowaniu etykiety starannie wycięto i przyklejono do kart, dbając o ich właściwe dopasowanie oraz estetykę wykonania. Dzięki temu każda karta zyskała wygląd profesjonalnego identyfikatora. Efekt końcowy został przedstawiony na fotografii na Rys. 3.6.

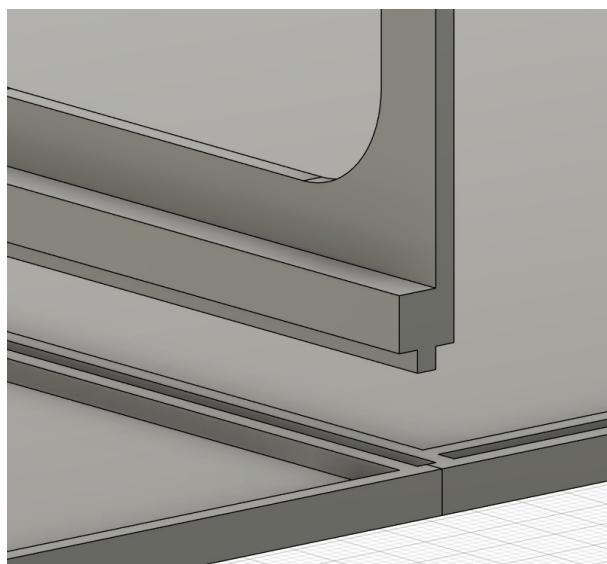


Rys. 3.6 Fotografia gotowych kart RFID z nałożoną grafiką

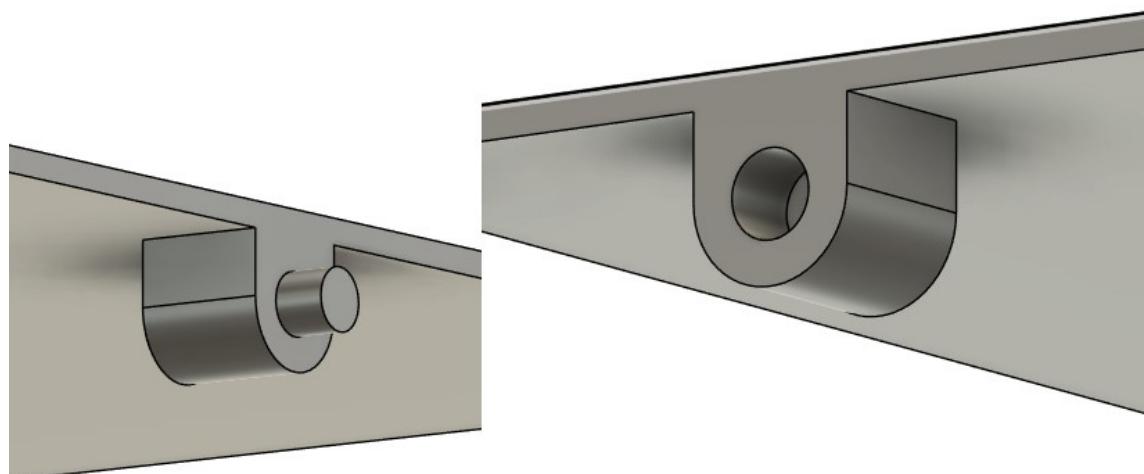
### 3.4. Zaprojektowanie obudowy i wydruk 3D

Proces projektowania obudowy rozpoczęto w programie Autodesk Fusion, uwzględniając zarówno rozmieszczenie komponentów elektronicznych, jak i wymiary fizyczne maszyny oraz jej estetykę. Ze względu na ograniczoną przestrzeń roboczą drukarki 3D *Ender 3 V2*, cały model musiał zostać podzielony na szereg mniejszych segmentów. Każdy z tych elementów został zaprojektowany tak, aby możliwe było jego łatwe i precyzyjne połączenie z pozostałymi częściami.

W tym celu opracowano dedykowane rozwiązania montażowe, takie jak prowadnice (Rys. 3.7), gniazda z pinami (Rys. 3.8) czy otwory pozycjonujące. Dzięki nim łączenie poszczególnych fragmentów obudowy odbywało się intuicyjnie i bez konieczności stosowania dodatkowych narzędzi.

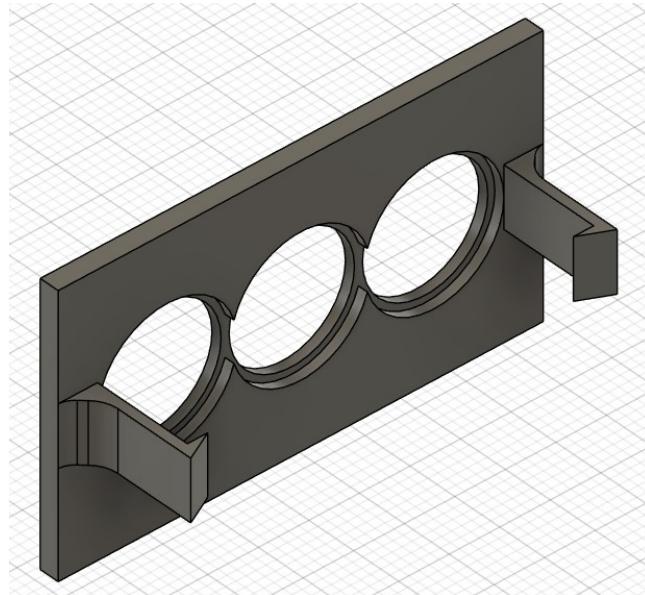


Rys. 3.7 Prowadnica łącząca ze sobą dwa elementy

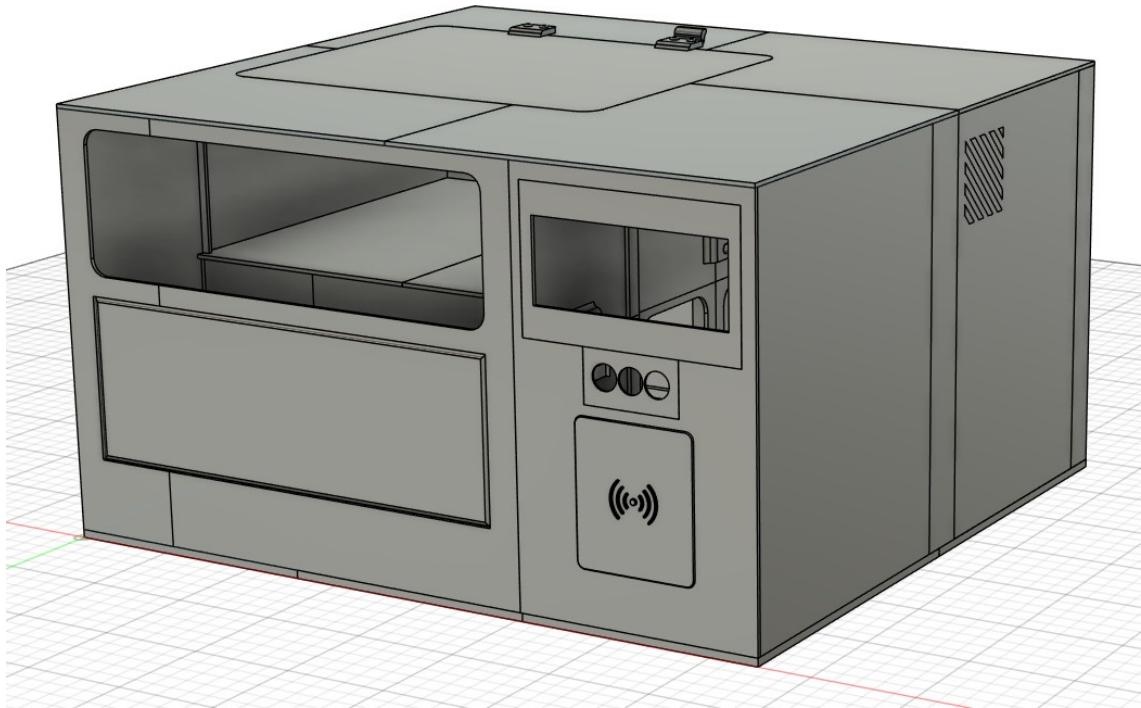


Rys. 3.8 gniazdo z pinem umożliwiające połączenie dwóch elementów podłoża

Niektóre komponenty takie jak otwory na przyciski, adaptery do sprężyn czy inne elementy wymagające częstych modyfikacji — zostały zaprojektowane jako osobne obiekty. Takie podejście znaczaco ułatwiało proces ich prototypowania oraz testowania. Mimo to, finalna integracja tych elementów z głównym modelem obudowy została zautomatyzowana, co pozwoliło zachować spójność konstrukcji i przyspieszyć końcowy etap projektowania. Na Rys. 3.9 został przedstawiony model otworów na przyciski, natomiast Rys. 3.10 przedstawia skończony model obudowy maszyny.



Rys. 3.9 Otwory na przyciski w programie Autodesk Fusion

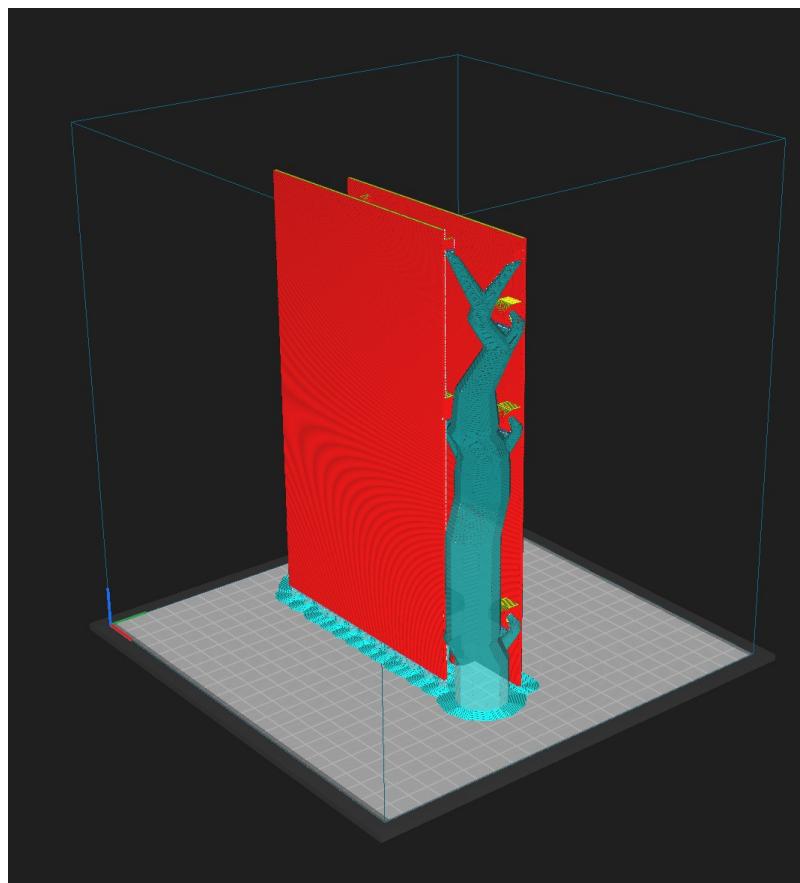


Rys. 3.10 Obudowa mini maszyny vendingowej

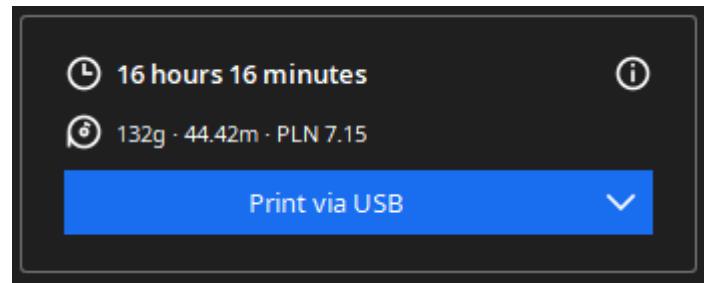
Po zakończeniu projektowania wszystkich elementów obudowy przystąpiono do ich fizycznego wytwarzania za pomocą drukarki 3D *Ender 3 V2*. Modele przygotowano w *slicerze Cura*, gdzie nadano im odpowiednie parametry druku dostosowane do wybranych materiałów oraz ograniczeń sprzętowych, a następnie dokonano ich przetworzenia na warstwy (ang. *slicing*), co umożliwiło właściwą interpretację przez drukarkę.

W trakcie pierwszych prób pojawiły się trudności związane ze słabą przyczepnością modeli do stołu roboczego, co skutkowało ich odrywaniem się podczas drukowania, zwłaszcza przy większych elementach podstawy. W celu wyeliminowania tego problemu skorzystano z funkcji poprawiających przyczepność dostępnych w oprogramowaniu Cura. Zastosowano m.in. opcję *brim*, która dodaje dookoła modelu cienką ramkę zwiększającą powierzchnię kontaktu z platformą. Dodatkowo, pierwsza warstwa została ustawiona jako nieco grubsza niż pozostałe, co poprawiło jej przyleganie do stołu i zredukuowało ryzyko przesunięcia się wydruku.

Na Rys. 3.11 przedstawiono dwa elementy podstawy maszyny po przetworzeniu na warstwy w programie Cura. Z kolei Rys. 3.12 prezentuje podsumowanie tego samego wydruku – zawiera informacje o przewidywanym czasie realizacji oraz zużyciu filamentu, co pozwoliło na lepsze planowanie i oszacowanie kosztów całej produkcji.



Rys. 3.11 Dwa elementy maszyny podzielone na warstwy z widoczną strukturą podporową



Rys. 3.12 Podsumowanie wydruku przedstawiające czas wydruku i zużyte zasoby

Na kolejnym zdjęciu (Rys. 3.13) przedstawiono adaptery sprężyn podczas procesu drukowania, co pozwala zobaczyć ich szczegóły oraz sposób wykonania. Natomiast na Rys. 3.14 widoczny jest finalny wydruk panelu frontowego, na którym zamontowano przyciski oraz wyświetlacz LCD, prezentując gotowy do montażu element urządzenia.



Rys. 3.13 AdAPTERY sprężyn w trakcie drukowania



Rys. 3.14 Frontowy panel maszyny vendingowej

### **3.5. Montaż urządzenia i testy końcowe**

Montaż urządzenia przebiegał z wykorzystaniem wcześniej zaprojektowanych i wydrukowanych elementów konstrukcyjnych, takich jak prowadnice oraz specjalne otwory montażowe. Dzięki nim połączenie poszczególnych części było znacznie łatwiejsze i precyzyjniejsze, co znacznie usprawniło cały proces składania maszyny.

Do trwałego łączenia elementów plastikowych zastosowano klej epoksydowy, który zapewnia wyjątkowo mocne i odporne połączenie. Dzięki temu konstrukcja jest stabilna i wytrzymała, co jest szczególnie istotne w kontekście codziennej eksploatacji urządzenia.

Po zakończeniu montażu przystąpiono do testowania urządzenia w celu weryfikacji poprawności działania wszystkich jego komponentów. Sprawdzono funkcjonalność silników krokowych, poprawność odczytu kart RFID oraz prawidłowe wyświetlanie informacji na ekranie LCD. Testy obejmowały również działanie przycisków i brzęczyka. Dzięki kompleksowej diagnostyce możliwe było szybkie wykrycie i usunięcie ewentualnych usterek, co zapewniło stabilną i niezawodną pracę całego systemu.

## **4. NAJWAŻNIEJSZE ELEMENTY OPROGRAMOWANIA**

Projekt został zrealizowany w języku C++ z podziałem na moduły odpowiedzialne za obsługę kluczowych komponentów urządzenia. W niniejszym rozdziale przedstawiono najistotniejsze fragmenty kodu, które odpowiadają za realizację podstawowych funkcji systemu: zarządzanie saldem użytkownika, wybór i zakup produktów, obsługę silników odpowiedzialnych za wydawanie towaru oraz implementację logiki głównej programu. Całość kodu źródłowego programu można znaleźć w repozytorium na GitHubie: [github.com/Atyxon/mini-vending-machine](https://github.com/Atyxon/mini-vending-machine).

## 4.1. Pętla główna i logika transakcji

Centralnym elementem programu jest funkcja pętli głównej *loop()*, która odpowiada za nieprzerwane monitorowanie interakcji użytkownika z systemem. Analizowane są wciśnięcia przycisków wyboru, sprawdzany jest stan dostępnych środków oraz dostępność wybranego produktu. W przypadku spełnienia warunków zakupu aktywowany zostaje odpowiedni mechanizm wydający produkt, a z salda użytkownika odejmowana jest kwota odpowiadająca cenie wybranego towaru. Fragment kodu prezentujący powyższe operacje został przedstawiony na Rys. 4.1

```
void loop() {
    int buttonState = buttonHandler.ButtonListen(); // Wait for user input
    displayHandler.processTimer();

    Serial.println(buttonState);
    if(buttonState != lastButtonState && buttonState != 0) { // Prevent button hold from registering as multiple presses
        lastButtonState = buttonState;
        selectedItem = buttonState-1;

        displayHandler.displaySelectMenu(items[selectedItem].name, items[selectedItem].cost);
        buzzer.ButtonBuzz();
    }

    // Reset the loop if no new card present on the sensor/reader. This saves the entire process when idle.
    if (!mfrc522.PICC_IsNewCardPresent() == false) {
        return;
    }

    // Select one of the cards
    if (!mfrc522.PICC_ReadCardSerial() == false) {
        return;
    }

    // Store UID in a temporary variable (String)
    String uidString = "";
    for (byte i = 0; i < mfrc522.uid.size; i++) {
        // Append a leading zero if the value is less than 0x10 for better formatting
        if (mfrc522.uid.uidByte[i] < 0x10) {
            uidString += "0";
        }
        uidString += String(mfrc522.uid.uidByte[i], HEX);
        if (i < mfrc522.uid.size - 1) {
            uidString += " ";
        }
    }

    ProcessCardAction(uidString);

    if (displayHandler.timerValue > 0)
    {
        selectedItem = ITEM_NOT_SELECTED;
    }

    // Halt the card to prevent reading the same one multiple times
    mfrc522.PICC_HaltA();
}
```

Rys. 4.1 Główna pętla programu

## 4.2. Zarządzanie kredytami użytkownika

System zarządza saldem użytkownika, które jest inicjalizowane i modyfikowane wyłącznie na poziomie oprogramowania. Saldo przechowywane jest w zmiennoprzecinkowych i aktualizowane w ramach logiki programu, zgodnie z przebiegiem transakcji i realizowanych operacji. Na Rys. 4.2 zostały przedstawione metody służące do zarządzania saldem użytkownika.

```
// Method to add credits
void addCredits(float amount) {
    if (amount < 0)
        return;
    credits += amount;
}

// Method to deduct credits
bool deductCredits(float amount) {
    if (amount < 0)
        return false;
    if (credits >= amount) {
        credits -= amount;
        return true;
    }
    return false; // Not enough credits
}
```

Rys. 4.2 Metody dodające oraz odejmujące kredyty

## 4.3. Zarządzanie produktami

Każdy obiekt reprezentujący produkt zawiera podstawowe informacje, takie jak nazwa oraz cena. System weryfikuje, czy użytkownik dysponuje wystarczającą ilością kredytów na dokonanie zakupu. Klasa *Item* została pokazana na Rys. 4.3

```
class Item {
public:
    int id;
    String name;
    float cost;

    // Constructor
    Item(int _id, String _name, float _cost) {
        id = _id;
        name = _name;
        cost = _cost;
    }
};
```

Rys. 4.3 Klasa Item

#### 4.4. Wydawanie produktów

Za fizyczne wydanie towaru odpowiadają silniki krokowe, które uruchamiane są poprzez przypisane im piny mikrokontrolera. Po zatwierdzeniu zakupu silnik wykonuje obrót sprężyny, która mechanicznie wypycha wybrany produkt z automatu. Fragment kodu ukazujący w jaki sposób są wysyłane sygnały do cewek silnika krokkowego, został zaprezentowany na Rys 4.4

```
void WritePins(int valPin1, int valPin2, int valPin3, int valPin4)
{
    pcf.write(pin1, valPin1);
    pcf.write(pin2, valPin2);
    pcf.write(pin3, valPin3);
    pcf.write(pin4, valPin4);
}

void OneStep(bool dir)
{
    if(dir)
    {
        switch[step_number]
        {
            case 0:
                WritePins(1,0,0,0);
                break;
            case 1:
                WritePins(0,1,0,0);
                break;
            case 2:
                WritePins(0,0,1,0);
                break;
            case 3:
                WritePins(0,0,0,1);
                break;
        }
    }else{
        switch(step_number){
        case 0:
            WritePins(0,0,0,1);
            break;
        case 1:
            WritePins(0,0,1,0);
            break;
        case 2:
            WritePins(0,1,0,0);
            break;
        case 3:
            WritePins(1,0,0,0);
        }
    }
    step_number = (step_number + 1) % 4;
}
```

Rys. 4.4 Fragment kodu odpowiedzialny za wysyłanie sygnału do cewek silnika krokkowego

## 5. PODSUMOWANIE

Celem niniejszej pracy było zaprojektowanie, wykonanie oraz oprogramowanie mini maszyny *vendingowej* sterowanej przy pomocy mikrokontrolera Arduino. Realizacja projektu umożliwiła praktyczne zastosowanie zdobytej wiedzy z zakresu systemów wbudowanych, elektroniki, automatyki oraz technologii druku 3D.

### 5.1. Osiągnięte cele projektu

W ramach projektu udało się zbudować w pełni działający prototyp automatu vendingowego sterowanego mikrokontrolerem którego fotografia została uwieczniona na Rys. 5.1. Urządzenie umożliwia wybór produktów, korzystanie z wewnętrznego systemu kredytowego oraz wydawanie towaru.



Rys. 5.1 Wygląd końcowy mini maszyny *vendingowej*

Integralną częścią pracy było również zaprojektowanie i wykonanie obudowy urządzenia z wykorzystaniem programu Autodesk Fusion oraz technologii druku 3D. Dzięki temu powstał w pełni funkcjonalny, fizyczny prototyp, który może być wykorzystany zarówno w celach edukacyjnych, jak i jako baza do dalszego rozwoju.

Osiągnięto:

- wykonanie obudowy i mechaniki,
- stworzenie oprogramowania sterującego,
- wdrożenie systemu kredytowego,
- pozytywne testy działania całego systemu.

## 5.2. Możliwości rozwoju i usprawnienia projektu

Jednym z istotnych kierunków rozwoju projektu jest integracja automatu ze zdalną bazą danych, obsługiwana z poziomu strony internetowej. W połączeniu z modułem Wi-Fi umożliwiły to bezprzewodowe zarządzanie urządzeniem, zdalne monitorowanie stanu kont użytkowników, dostępności przedmiotów, a także aktualizację danych i konfiguracji w czasie rzeczywistym. Takie rozwiązanie znaczaco zwiększyły funkcjonalność oraz ułatwiło obsługę i administrację automatem.

Dodatkowo, warto rozważyć rozbudowę systemu o większą liczbę slotów na przedmioty, co pozwoliłoby na oferowanie szerszego asortymentu oraz lepsze dopasowanie automatu do różnych zastosowań.

Kolejnym możliwym usprawnieniem projektu jest modyfikacja automatu w taki sposób, aby umożliwiał codzienne wydawanie określonych przedmiotów, na przykład jednej pary rękawiczek pracownikom zakładu, co jednocześnie służyłoby jako forma rejestracji obecności pracownika w pracy. Dzięki takiemu rozwiązaniu możliwe byłoby automatyczne prowadzenie ewidencji czasu pracy oraz kontrola zużycia materiałów ochronnych, co mogłoby przynieść wymierne korzyści zarówno w zakresie logistyki, jak i organizacji pracy.

### **5.3. Podsumowanie kosztów projektu**

W trakcie realizacji projektu szczególną uwagę zwrócono na kontrolę kosztów związanych z zakupem komponentów elektronicznych, materiałów do druku 3D oraz innych niezbędnych akcesoriów. Celem było utrzymanie wydatków na rozsądny poziomie, przy jednoczesnym zapewnieniu wysokiej jakości i funkcjonalności urządzenia. Wszystkie elementy zostały starannie dobrane pod kątem dostępności oraz ceny, co pozwoliło zoptymalizować budżet projektu.

W tabeli poniżej przedstawiono szczegółowe zestawienie kosztów poszczególnych części oraz materiałów wykorzystanych w projekcie.

Nazwa komponentu	Ilość	Łączny koszt [PLN]
Arduino 101	1	190
Wyświetlacz LCD 2004	1	18
Czytnik RFID RC522	1	10
Silnik krokowy 28BYJ-48	3	30
Magistrala PCF8574	2	13
Sprężyny	3	96
Tranzystor	1	3
Zestaw przewodów	1	15
Zestaw rezistorów	1	17
Zestaw kart RFID	1	9
Zestaw przycisków	1	8
Brzęczyk	1	4
Zasilacz	1	14
Filament do drukarki 3D	1	91
Klej epoksydowy	2	12
Naklejka na karty RFID	1	10

Podsumowując, całkowite koszty związane z realizacją projektu, obejmujące zakup komponentów elektronicznych, filamentu do druku 3D oraz dodatkowych elementów konstrukcyjnych, wyniosły około 540 zł.

### **5.4. Wnioski końcowe**

Wnioski płynące z realizacji projektu wskazują, że budowa własnych urządzeń *vendingowych* z użyciem otwartych platform sprzętowych jest możliwa przy stosunkowo niskich kosztach oraz dużej elastyczności projektowej. Projekt udowodnił, że nawet niewielkie systemy wbudowane mogą realizować złożone funkcje logiczne i mechaniczne, co czyni je bardzo użytecznymi w dziedzinie automatyki, prototypowania i edukacji technicznej.

## BIBLIOGRAFIA

1. Paul Scherz, Practical Electronics for Inventors, 2016
2. Richard Horne, 3D Printing For Dummies, 2023
3. Jeremy Blum, Exploring Arduino: Tools and Techniques for Engineering Wizardry, 2013
4. [https://www.mouser.com/datasheet/2/758/stepd-01-data-sheet-1143075.pdf?  
srsltid=AfmBOorIFhN\\_tfQyGc5Wqmq1L0cBsYLBVj2G1f2\\_evs0ksS9Y87X0dG](https://www.mouser.com/datasheet/2/758/stepd-01-data-sheet-1143075.pdf?srsltid=AfmBOorIFhN_tfQyGc5Wqmq1L0cBsYLBVj2G1f2_evs0ksS9Y87X0dG)
5. <https://www.hadex.cz/navody/m490a.pdf>
6. [https://www.nxp.com/docs/en/data-sheet/PCF8574\\_PCF8574A.pdf](https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf)
7. <https://myduino.com/product/12x12mm-tactile-push-button/>
8. [https://uk.beta-layout.com/download/rk/RK-10290\\_410.pdf](https://uk.beta-layout.com/download/rk/RK-10290_410.pdf)
9. Tutorial books, Autodesk Fusion 360 For Beginners, 2021
10. John Boxall, Arduino Workshop, 2021
11. Bjarne Stroustrup, The C++ Programming Language, 2013
12. Scott Meyers, Effective Modern C++, 2014
13. <https://docs.arduino.cc/retired/boards/arduino-101-619/>

## SPIS ILUSTRACJI

Rys. 1.1 Grafika koncepcyjna.....	7
Rys. 2.1 Starożytna maszyna wydająca świętą wodę.....	10
Rys. 2.2 Płytki Arduino 101.....	11
Rys. 2.3 Silnik krokowy 28BYJ-48.....	12
Rys. 2.4 Moduł RFID RC522.....	13
Rys. 2.5 Magistrala I <sup>2</sup> C PCF8574.....	14
Rys. 2.6 Wyświetlacz LCD 4x20.....	15
Rys. 2.7 Okno projektu w programie Autodesk Fusion.....	16
Rys. 3.1 Komunikat oznajmiający o nieznanym identyfikatorze karty.....	19
Rys. 3.2 Dwie magistrale PCF8574 z różną konfiguracją zworek.....	20
Rys. 3.3 Przedstawienie działania tranzystora w projekcie.....	21
Rys. 3.4 Graficzna wizualizacja połączeń w projekcie.....	22
Rys. 3.5 Przedstawienie projektu graficznego karty.....	23
Rys. 3.6 Fotografia gotowych kart RFID z nałożoną grafiką.....	24
Rys. 3.7 Prowadnica łącząca ze sobą dwa elementy.....	25
Rys. 3.8 gniazdo z pinem umożliwiające połączenie dwóch elementów podłożą.....	25
Rys. 3.9 Otwory na przyciski w programie Autodesk Fusion.....	26
Rys. 3.10 Obudowa mini maszyny <i>vendingowej</i> .....	26
Rys. 3.11 Dwa elementy maszyny podzielone na warstwy z widoczną strukturą podporową.....	27
Rys. 3.12 Podsumowanie wydruku przedstawiające czas wydruku i zużyte zasoby.....	28
Rys. 3.13 Adaptery sprężyn w trakcie drukowania.....	28
Rys. 3.14 Frontowy panel maszyny <i>vendingowej</i> .....	29
Rys. 4.1 Główna pętla programu.....	31
Rys. 4.2 Metody dodające oraz odejmujące kredyty.....	32

Rys. 4.3 Klasa Item.....	32
Rys. 4.4 Fragment kodu odpowiedzialny za wysyłanie sygnału do cewek silnika krokowego.....	33
Rys. 5.1 Wygląd końcowy mini maszyny <i>vendingowej</i> .....	34