



Instytut Informatyki Politechniki Śląskiej
Zespół Mikroinformatyki i Teorii Automatów
Cyfrowych

**Rok akademicki:**

Rodzaj studiów*: SSI/NSI/NSM

Przedmiot (Języki Asemblerowe/SMiW):

Grupa

Sekcja

2020/2021

SSI

SMIW

3

1

Imię:

Michał

Prowadzący:

JP

Nazwisko:

Urbanek

OA/JP/KT/GD/BSz/GB

Raport końcowy

Temat projektu:

Gra retro - Tetris

Data oddania:
dd/mm/rrrr

27.01.2021

1.Opis założeń projektu



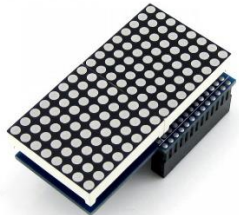
Projektowanym układem jest gra Tetris typu retro. Składają się na nią połączone ze sobą matryce LED, na których wyświetlane są spadające bloki. Kształt bloków jest generowany losowo. Interakcja z użytkownikiem odbywa się za pomocą przycisków. Na dodatkowym wyświetlaczu prezentowane są punkty zdobyte przez gracza. Gra kończy się w momencie wypełnienia wyświetlacza w pionie. Podczas rozgrywki występują efekty dźwiękowe oraz melodie. Zasilanie odbywa się za pomocą powerbanka lub innego źródła zasilania 5V

2. Analiza i wybór elementów elektronicznych

Gra będzie posiadać wyświetlacz w postaci małej matrycy LED (np. 2x8x8pix). Interakcja będzie odbywać się za pomocą małych przycisków tact-switch z nakładkami. Punkty zdobyte przez gracza będą wyświetlane na dodatkowym wyświetlaczu 7-segmentowym x 4 cyfry. Efekty dźwiękowe będą realizowane przez brzęczyk (buzzer). Układ będzie zasilany z 5V źródła – z powerbanka.

Wybór i uzasadnienie: Decyzja jest uzasadniona chęcią posiadania urządzenia mobilnego (gra w stylu retro), a sposób ten proponuje kompaktowe urządzenie. Zastosowanie gotowej, małej matrycy LED jest optymalnym rozwiązaniem z punktu poboru energii oraz rozmiarów. Dodatkowy wyświetlacz 7-segmentowy x 4 cyfry będzie konieczny do wyświetlania punktów, sam wyświetlacz również jest kompaktowy. Zastosowany brzęczyk jest kompaktowy oraz w pełni funkcjonalny. Zasilanie 5V z Powerbank'a, eliminuje problem ciągłego rozładowywania baterii – jest rozwiązaniem ekologicznym. Dodatkowo zapewnia prosty sposób ponownego ładowania akumulatora.


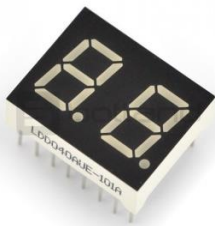
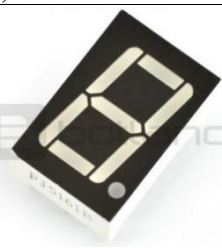
1. Matryca LED

	LED 8x8 + MAX7219	4xLED 8x8 + MAX7219	Matryca LED 16x8 MAX7219 - Waveshare 9862
Typ matrycy	LED	LED	LED
Przekątna	1,77"	4x1,77"	2x1,77"
Interfejs	GPIO	GPIO	GPIO
Rozdzielczość	8x8pix	4x8x8pix	2x8x8pix
Napięcie zasilania	5V	5V	5V
Dodatkowy sterownik	Tak	Nie	TAK
Wymiary	32x32mm	128x32mm	64,5 x 35 x 26,5 mm
Interfejs komunikacji	SPI	SPI	SPI
Kolor diody	Czerwony	Czerwony	RGB
Dodatkowy zestaw	TAK	TAK	NIE
Cena	12,90zł	26,25zł	29,90zł
Zdjęcie poglądowe			
Wybór	✗	✗	✓

Uzasadnienie wyboru: Matryca LED 16x8 Waveshare 9862 jest bardzo kompaktowym układem. Posiada odpowiednie napięcie pracy, interfejs komunikacyjny oraz jest bardzo popularnym i sprawdzonym układem (jest dedykowany urządzeniom Raspberry PI). Układ dodatkowo posiada sterownik oraz jest w całości zmontowany i polutowany. Jest wyposażony w obszerną dokumentację oraz opcjonalnie umożliwia montaż na przedniej ścianie obudowy (z wykorzystaniem taśmy transmisyjnej)




2. Wyświetlacz 7-segmentowy

	Wyświetlacz LED 7-seg. 3 cyfry	Wyświetlacz 7-seg. 2 cyfry	Wyświetlacz 7-seg. 1 cyfra
Ilość znaków	4	2	1
Kolor wyświetlacza	Czerwony	Czerwony	Czerwony

Kolor tła	Czarny	Czarny	Czarny
Wymiary	50,3 x 19 mm	20x16mm	12x19
Jasność	5-10mcd	6,5mcd	5-10mcd
Prąd pracy	5-10mA, max 20mA	10mA	0,3A
Rodzaj	Wspólna anoda	Wspólna anoda	Wspólna anoda
Cena	3,90zł	2,30zł	1,10zł
Zdjęcie poglądowe			
	✓	✗	✗

Uzasadnienie wyboru: Wyświetlacz LED 7-seg. 4 cyfry jest najbardziej odpowiednim wyborem, ze względu na posiadanie aż 4 cyfr umieszczonych w jednej obudowie. Jest to wygodniejsze rozwiązanie, niż zastosowanie 4 wyświetlaczy 7 segmentowych. Czerwony kolor wyświetlanych cyfr będzie dobrze komponować się z czerwonym kolorem matrycy LED. Pobór prądu oraz jasność również są na rozsądnym poziomie.



3. Mikrokontroler

	Mikrokontroler AVR - ATmega328P-U DIP	Mikrokontroler AVR - ATmega16A-AU SMD	Mikrokontroler AVR - ATmega88PA-PU DIP
Zasilanie	1,8V-5,5V	4,5-5,5V	2,7-5,5V
Taktowanie	Do 20MHz	16MHz	Do 20 MHz
Pamięć Flash	32KB	16KB	8KB
Linie wejścia/wyjścia	23	32	23
Interfejsy komunikacyjne	USART/SPI/TWI(I2C)	USART/SPI	USART/SPI/I2C
Licznik 8/16bit	2/1	2/1	2/1
Kanały A/C	6	8	1
Kanały PWM	6	4	?
Rodzaj montażu	THT	SMD	THT
Cena	12,90zł	8,90zł	12,90zł
Zdjęcie poglądowe			
Wybór	✓	✗	✗

Uzasadnienie wyboru: Mikrokontroler AVR - ATmega328P-U DIP jest bardzo dobrym wyborem w proponowanym układzie elektronicznym. Posiada wszystkie niezbędne parametry, tj. taktowanie czy napięcie pracy oraz linie wejścia/wyjścia czy liczniki. Jest wyposażony we wszystkie najważniejsze interfejsy komunikacyjne. Dodatkowo układ jest niezwykle popularny oraz posiada obszerną dokumentację. Mikrokontroler jest montowany w technologii THT oraz istnieje możliwość wlutowania gniazda na mikrokontroler – co może być przydatne w przypadku uszkodzenia układu (wystarczy wyjąć układ z gniazda i wsadzić nowy). Jego cena również nie jest wysoka.




4. Brzęczyk (buzzer)

	Buzzer z generatorem 5V	Moduł z buzzerem aktywnym bez generatora
Napięcie zasilania	5V	5V
Głośność	85dB	85dB
Pobór prądu	30mA	30mA
Częstotliwość	2,3kHz +/- 500Hz	2,3kHz +/- 500Hz
Typ obudowy	THT	THT

Uwagi		Układ w postaci modułu, nie posiada wzmacniacza tranzystorowego
Cena	0,90zł	4,50zł
Zdjęcie poglądowe		
Wybór	✓	✗

Uzasadnienie wyboru: Buzzer z generatorem 5V jest bardzo dobrą opcją w proponowanym układzie. Posiada niewielkie wymiary, jest prosty w obsłudze przy czym spełnia swoją rolę. Napięcie zasilania jest dostosowane do reszty elementów układu, a głośność jest na odpowiednim poziomie. Element jest szeroko dostępny w sklepach oraz posiada niską cenę.

5. Przyciski

	Tact Switch 3x6mm / 5mm SMD	Tact Switch 6x6mm / 4,3mm THT	Tact Switch 12x12mm z nasadką
Rozmiar	3x6mm	6x6mm	12x12
Typ montażu	SMD	THT	THT
Kolor	Czarny	Czarny	Czarny
Typ przełącznika	Monostabilny	Monostabilny	Monostabilny
Wysokość	5mm	4.3mm	10mm
Ilość w zestawie	5szt	5szt	5
Cena	1,99zł	1,50zł	2,50zł
Zdjęcie poglądowe			
Wybór	✗	✗	✓

Uzasadnienie wyboru: Tact Switch 12x12mm z nasadką jest bardzo dobrym wyborem w proponowanym układzie. Posiada niewielkie rozmiary, jest montowany w technologii THT oraz posiada dodatkową nasadkę, co zwiększy komfort użytkowania. Jego cena jest najwyższa w całym zestawieniu, ale w dalszym ciągu bardzo niska.

Lista narzędzi i materiałów użytych w projekcie:

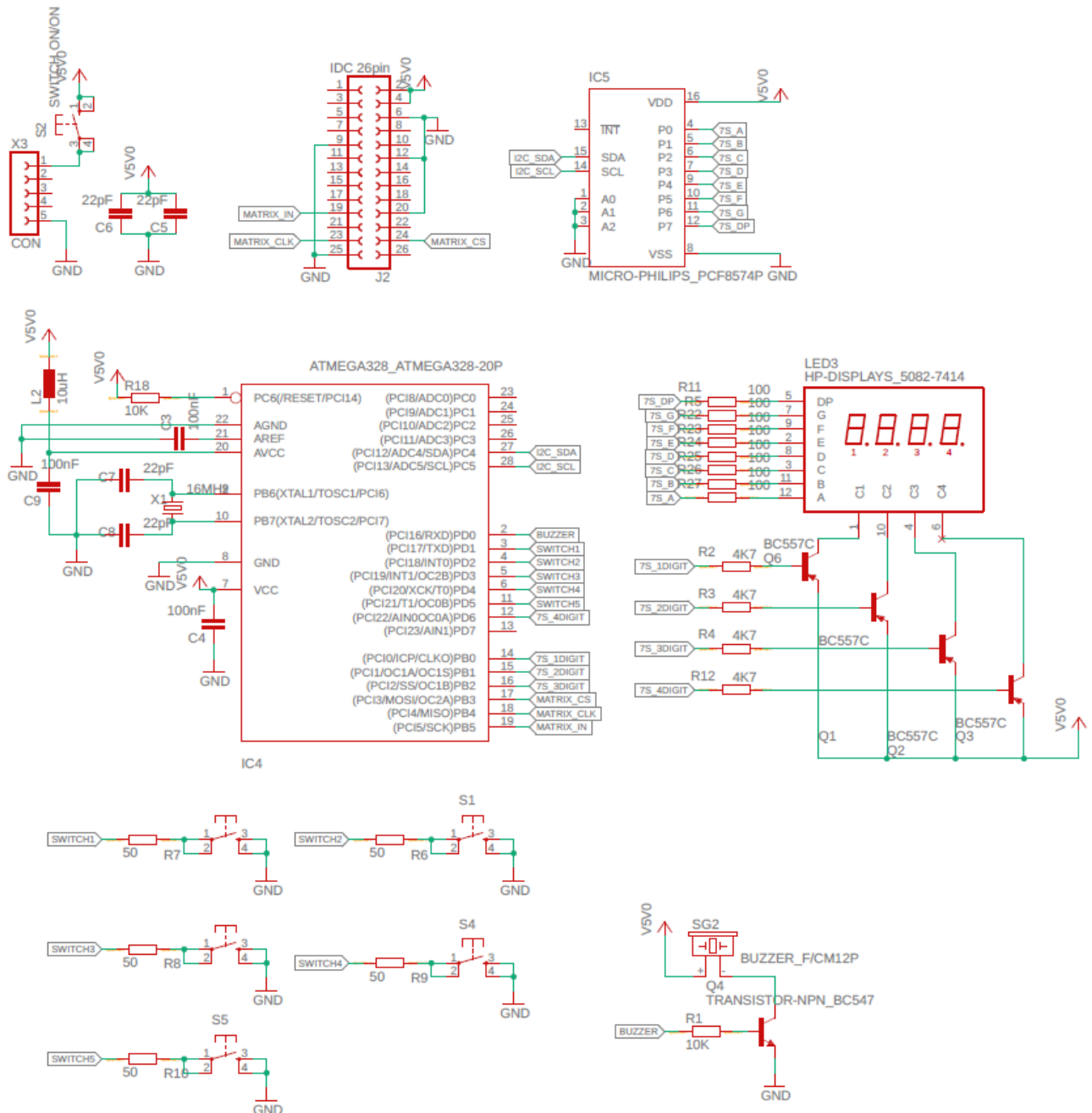
Lutownica	ZD-20U Mini lutownica USB ZD20U 8W 5V
Cyna z topnikiem	POIWO LUTOWNICZE FIOŁKA LC60 10g
Zestaw wkrętaków	ZESTAW PRECYZYJNYCH ŚRUBOKRĘTÓW 25W1 TORX WKRĘTAKI
Cążki boczne do przewodów	Velleman VT301
Kalafonia	Kalafonia aktywna 40g AG CHEMIA

3. Specyfikacja wewnętrzna

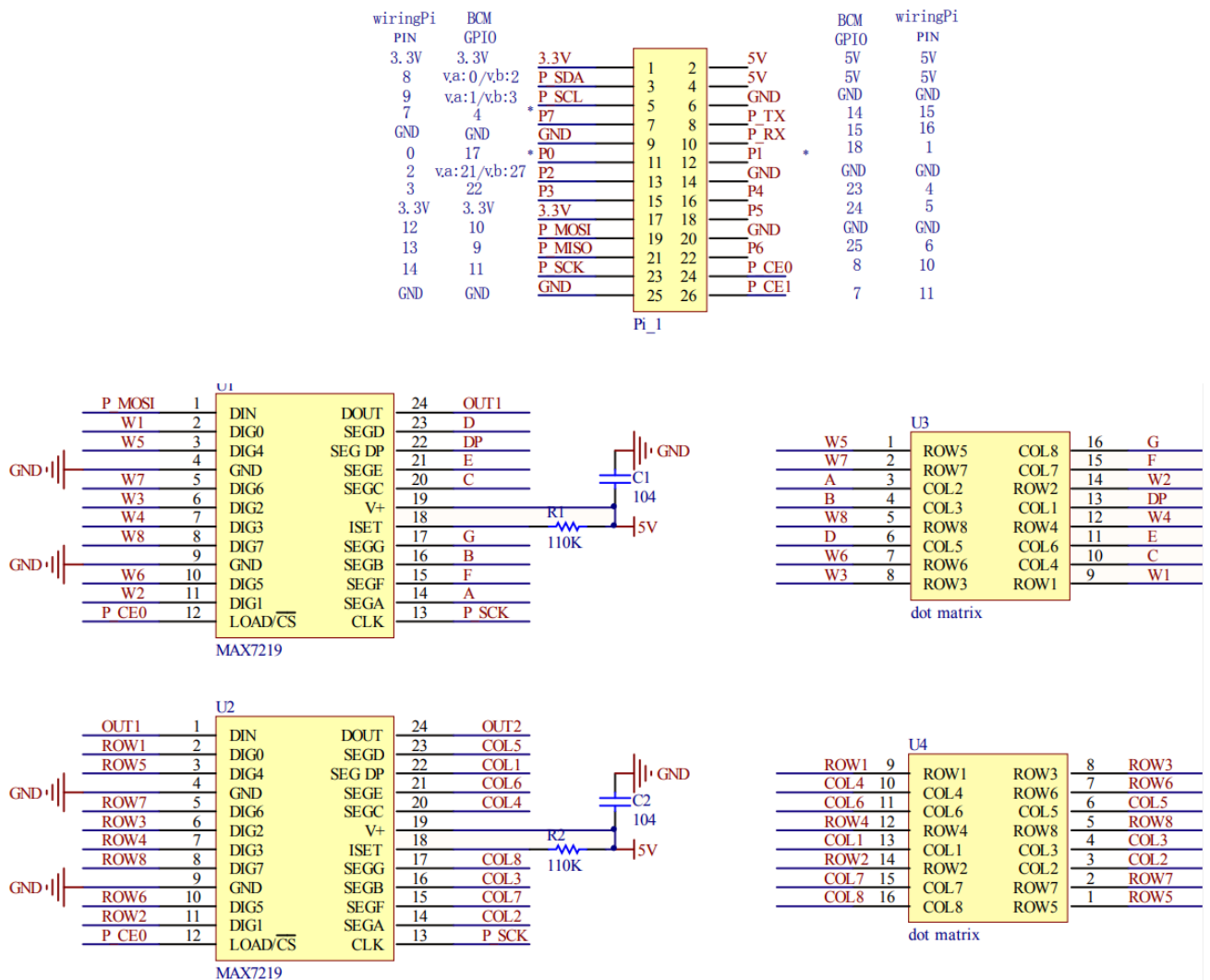
3.1 Schematy urządzenia

3.1.1 Schemat ideowy

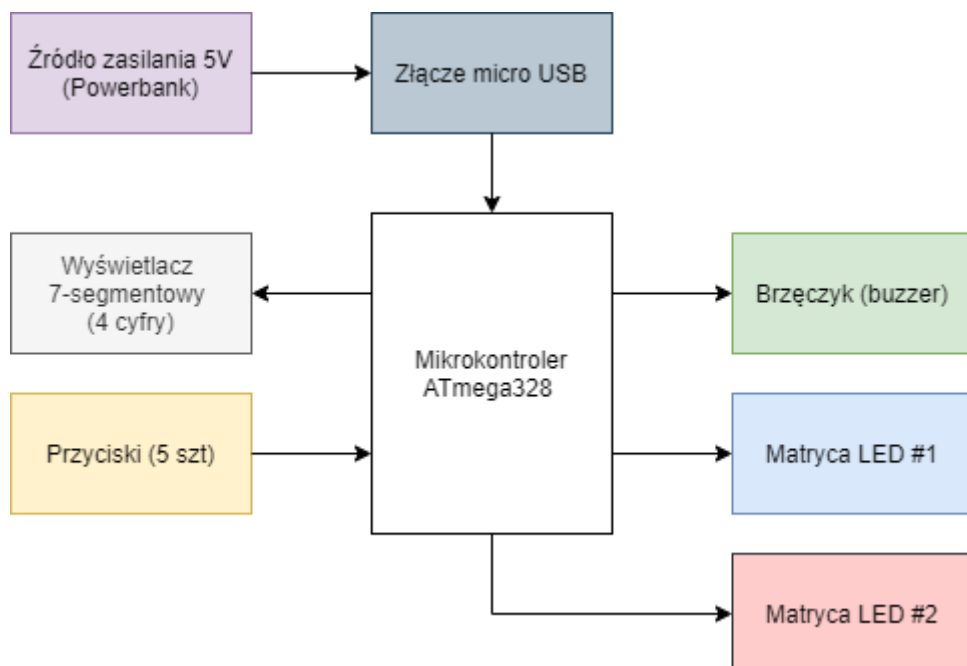
3.1.1.a Schemat ideowy układu



3.1.1.b Schemat ideowy matrycy Waveshare



3.1.2 Schemat blokowy



3.2 Opis funkcji poszczególnych bloków układu

3.2.1 Mikrokontroler

Jest to główny element układu. Jego zadaniem jest komunikacja z pozostałymi urządzeniami oraz zarządzanie nimi. Realizuje on program umieszczony w pamięci.

3.2.2 Wyświetlacz 7-segmentowy (4 cyfry)

Jest to element zliczający punkty gracza w czasie rozgrywki. Jest to wyświetlacz składający się z 4 cyfry, czyli jest w stanie zliczyć do 9999 pkt. Za każde umieszczenie bloku są przyznawane punkty oraz za zlikwidowanie całego wiersza.

3.2.3 Matryca LED

Jest głównym elementem wyświetlającym grę. Składa się z dwóch matryc 8x8. Matryca jest skomunikowana z mikrokontrolerem przez układy wspomagające umieszczone na wspólnej płytce (jest to cały moduł)

3.2.4 Brzęczyk

Jego zadaniem jest generowanie dźwięków podczas rozgrywki (np. dźwięk rotacji bloku) oraz odtwarzanie muzyki startowej

3.2.5 Źródło zasilania (Powerbank)

Jest to 5V źródło zasilania, o napięciu stałym. Zasilania odbywa się za pomocą przewodu microUSB. Istnieje również możliwość zasilania z portu PC

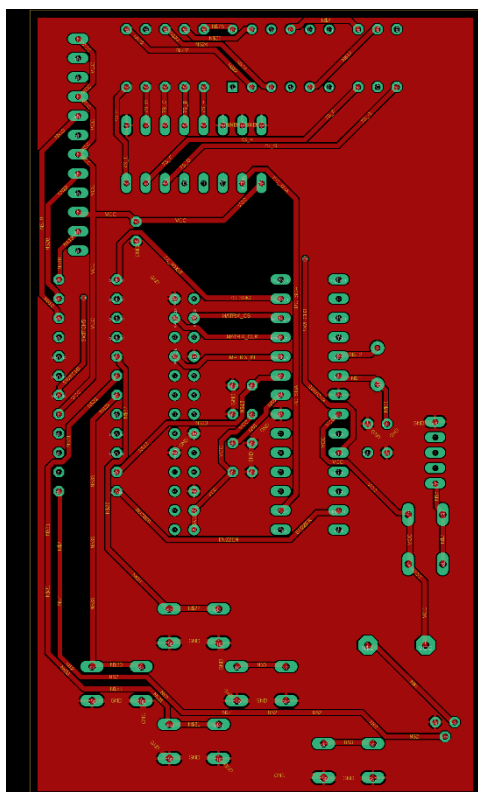
3.2.6 Przyciski

Przyciski są odpowiedzialne za interakcję użytkownika z grą. Występują przyciski: lewo, prawo, dół, rotacja, restart.

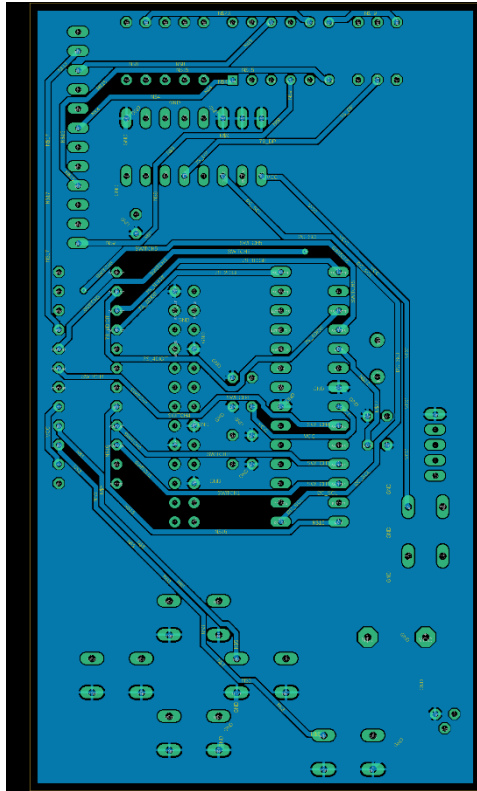
3.3 Projekt płytki drukowanej

3.3.1 Schemat płytki drukowanej

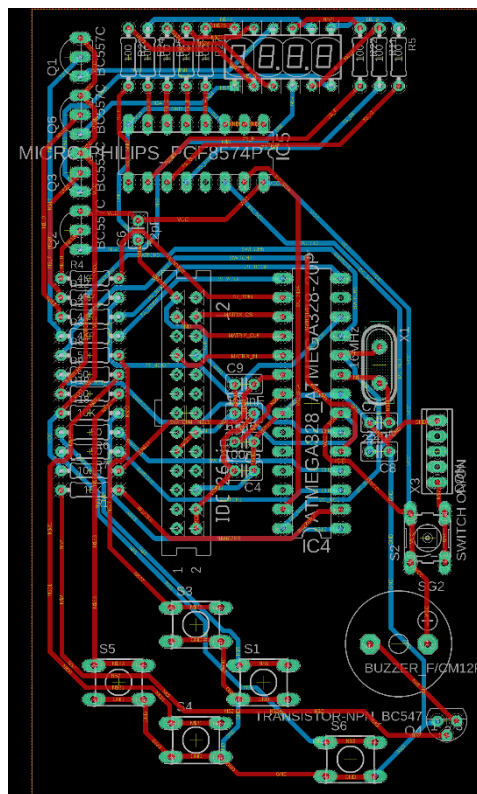
3.3.1.1 Góra płytki



3.3.1.2 Dół płytki



3.3.1.3 Obie strony płytki

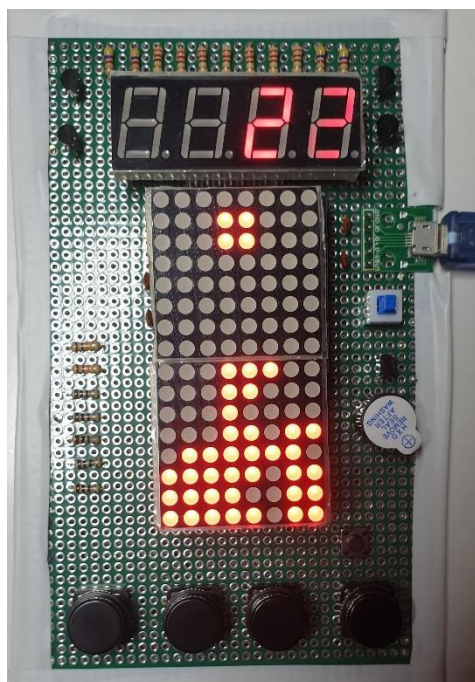


3.3.2 Rozmieszczenie elementów na płytce

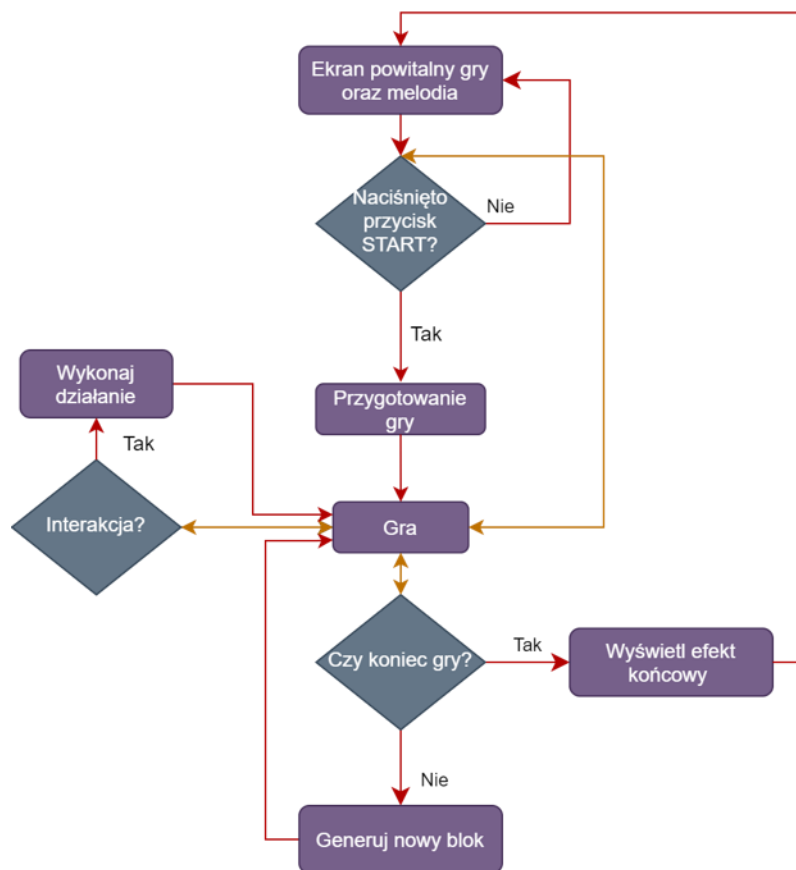
3.3.2.1 Front płytki – bez osadzonych elementów zewnętrznych



3.3.2.2 Front płytki – z osadzonymi elementami zewnętrznymi



3.4 Algorytm oprogramowania urządzenia



3.5 Opis wszystkich zmiennych

Nazwa zmiennej	Typ zmiennej	Opis
current_digit	byte	Zmienna przechowująca aktualną cyfrę licznika punktów
count	int	Zmienna przechowująca aktualną liczbę punktów gracza
lcd0	Int[]	Tablica zmiennych przechowująca aktualny wektor danych wyświetlanych na wyświetlaczu górnym
lcd1	Int[]	Tablica zmiennych przechowująca aktualny wektor danych wyświetlanych na wyświetlaczu dolnym
active	Long[]	Wektor danych przechowujący aktualne ruchomego ustawienie bloku
screen	Long[]	Wektor danych przechowujący aktualne ustawienie bloków na ekranie (nieruchomych bloków)
tempcol	Int []	Numer aktualnie przetwarzanego wiersza
figura	int	Numer figury- rodzaj bloku
figuranext	Int	Rodzaj następnej figury
fromleft	Int	Ustawienie nowego bloku od lewej
fromright	Int	Ustawienie nowego elementu od prawej
angle	Int	Kąt o jaki jest zrotowany blok
colcheck	Int	Sprawdzenie czy kolumna jest dostępna
movecheck	Int	Sprawdzenie czy żądany ruch jest

		dozwolony
score	Int	Licznik sumy punktów za wypełnienie wyświetlacza w poziomie
started	Int	Zmienna przechowująca wartość, czy gra została uruchomiona czy jest w menu głównym
lcrows	Int	Liczba wierszy w wyświetlaczu
lccols	Int	Liczba kolumn w wyświetlaczu
brickdelay	Int	Opóźnienie w opuszczaniu aktualnego bloku
level	Int	Poziom gry, im wyższy, tym bloki spadają z większą prędkością
sound	Bolean	Zmienna sprawdzająca czy ma być odtwarzany dźwięku menu startowego

Nazwa stałej	Numer pinu	Opis
SegA	4	Segment A wyświetlacza 7 segmentowego
SegB	5	Segment B wyświetlacza 7 segmentowego
SegC	6	Segment C wyświetlacza 7 segmentowego
SegD	7	Segment D wyświetlacza 7 segmentowego
SegE	8	Segment E wyświetlacza 7 segmentowego
SegF	9	Segment F wyświetlacza 7 segmentowego
SegG	10	Segment G wyświetlacza 7 segmentowego
Dig1	0	Cyfra nr 1 wyświetlacza 7 segmentowego
Dig2	1	Cyfra nr 2 wyświetlacza 7 segmentowego
Dig3	2	Cyfra nr 3 wyświetlacza 7 segmentowego
Dig4	3	Cyfra nr 4 wyświetlacza 7 segmentowego
rotate_button	3-expander	Przycisk rotacji bloków
left_button	2-expander	Przycisk przesunięcia bloku w lewo
right_button	0-expander	Przycisk przesunięcia bloku w prawo
start_button	4-expander	Przycisk uruchomienia gry od początku
speaker_pin	5-expander	Buzzer z generatorem
down_button	1-expander	Przycisk przesunięcia w dół

3.6 Opisz wszystkich funkcji i procedur

playTone()	Int Tone, int duration	Funkcja odpowiedzialna za odtwarzanie odpowiedniej nutki, z zadaną częstotliwością
playNote()	Char Note, int duration	Funkcja odtwarzająca wybrane melodie w odpowiedniej kolejności
updateColumn()	Int column	Funkcja aktualizująca wybraną kolumnę
ButtonDelay()	Int Bdelay	Funkcja wprowadzająca opóźnienie do wciskanych klawiszy
splashScreen()		Funkcja wyświetlająca na ekranie napisy startowe
Disp()	Byte numer	Funkcja wyświetlająca odpowiednią cyfrę na wyświetlaczu 7 segmentowym w dane chwili czasu
ISR(TIMER1_OVF_vect)		Procedura obsługi przerwania
Disp_off()		Funkcja wyłączająca wszystkie cyfry wyświetlacza 7-segmentowego w danej chwili czasu
Void Setup()	-	Funkcja zawierająca inicjalizację wszystkich niezbędnych portów, konfigurację liczników przerwań,

		inicjalizację wyświetlaczy, ustawienie jasności wyświetlaczy oraz ich pierwotnego wyczyszczenia
Rotation0()	-	Funkcja odpowiedzialna za rotowanie bloku o 0 stopni. Czyli rotacja nie jest wykonywana
Rotation90()	-	Funkcja odpowiedzialna za rotowanie bloku o 90 stopni
Rotation180()	-	Funkcja odpowiedzialna za rotowanie bloku o 180 stopni
Rotation270()	-	Funkcja odpowiedzialna za rotowanie bloku o 270 stopni
SetFigure()	Int[] Active, int figure	Funkcja ustawiająca wejściową tablicę na figure zadaną w parametrze
Reset()		Funkcja resetująca grę oraz przywracająca wartości zmiennych do ustawień domyślnych
gameOver()		Funkcja wyświetlająca końcową animację gry oraz przenosząca do menu startowego
Loop()		Główna funkcja gry będąca pętlą nieskończoną. Odbywają się tutaj następujące stany: rozpoczęcie gry, gra, koniec gry. Jest to funkcja zarządzająca wszystkimi pozostałymi takimi jak: funkcje generowania kształtów, przemieszczania bloków, rotacji, wydawania dźwięków, ogramiania melodii

4. Specyfikacja zewnętrzna

4.1 Opis funkcji elementów sterujących urządzeniem.

W układzie zostało zamontowane 5 przycisków. Przycisk: lewo- przesunięcie bloku w lewo, prawo- przesunięcie bloku w prawo, dół -szybsze przesunięcie bloku w dół, rotacja – rotowanie bloku oraz restart – gra zostaje uruchomiona od nowa. Wgrywanie aktualne oprogramowanie odbywa się przez wyjęcie mikrokontrolera u umieszczenie go w zewnętrznym programatorze – układ nie został wyposażony w złącze do programowania.

Układ jest wyposażony w przycisk bistabilny ON/OFF, jego zadaniem jest podanie do układu napięcia zasilania w pozycji ON .

4.2 Opis funkcji elementów wykonawczych

Wyświetlacz zamontowany w układzie jest odpowiedzialny za wyświetlanie aktualnego stanu gry. Zewnętrzny buzzer generuje dźwięki podczas gry oraz melodię startową. Wyświetlacz 7 segmentowy (4 cyfry) zlicza punkty gracza.

4.3 Opis reakcji oprogramowania na zdarzenia zewnętrzne

Gra reaguje na wciśnięte przez użytkownika przyciski. Wciśnięcie przycisku bistabilnego ON/OFF na pozycji ON powoduje uruchomienie gry. Naciśnięcie przycisku starty powoduje rozpoczęcie rozgrywki. Podczas gry do dyspozycji użytkownika są przyciski prawo ,lewo, dół, rotacja. Istnieje możliwość zresetowania urządzenia za pomocą przycisku restart. Wtedy punkty zostaną wyzerowane a gra zacznie się od początku

4.4 Skrócona instrukcja obsługi

1. Podłączyć źródło zasilania 5V do urządzenia za pomocą portu micro Usb.
2. Przełączyć przycisk bistabilny na pozycję ON w celu włączenia urządzenia
3. Używać przycisków nawigacyjnych w celu przemieszczania bloków
4. Nacisnąć przycisk Reset w celu rozpoczęcia gry od nowa.

5. Przyłączyć przycisk bistabilny na pozycję OFF w celu wyłączenia urządzenia

5. Opis montażu i uruchamiania:

W celu wykonania układu konieczne jest wykonanie odpowiedniej płytki lub wykorzystanie płytki uniwersalnej. Do zmontowania całości konieczna jest precyzyjna lutownica oraz cyna. W przypadku testowania układu możliwe jest modułowe wywoływanie funkcji

5.1 Elementy użyte w układzie wraz z kosztorysem

ELEMENT	CENA
Mikrokontroler ATmega328P-U	13.90 zł
Matryca LED 16x8 Waveshare 9862	29.90 zł
Ekspander wyprowadzeń PCF8574N	4.90 zł
Wyświetlacz 7 segmentowy 4 cyfry	3.90 zł
Moduł gniazda Micro USB	4.95 zł
Przyciski Tact-Switch (5szt)	2.95 zł
Płytki uniwersalna 150x90mm	11.00 zł
INNE (rezystory, kondensatory, diody, tranzystory, podstawki, przełączniki, rezonator, buzzer, przewody)	~40.00 zł
Koszt dostaw	25.00 zł
SUMA	~140 zł

5.2 Opis montażu układu oraz kolejnych testów.

1. Zakupić wszystkie niezbędne części oraz wykonać płytkę PCB lub zlecić jej wykonanie zewnętrznej firmie.
2. Wlutować wszystkie elementy w odpowiednich miejscach na płytce
 - a. Złącze gniazda mikrokontrolera
 - b. Rezystory, kondensatory oraz tranzystory
 - c. Złącza 26 pinowe wyświetlacza
 - d. 2 złącza 6 pinowe wyświetlacza 7 segmentowego
 - e. Złącze 16 pinowe ekspandera wyprowadzeń
 - f. Złącze micro usb
 - g. Buzzer z generatorem
 - h. 5 Przycisków tact-switch
 - i. Przycisk bistabilny ON/OFF
 - j. Rezonator kwarcowy 16MHZ
3. Po zmontowaniu układu należy wykonać test ciągłości elektrycznej (np. za pomocą multimetra). Pomiary te należy przeprowadzić przy wyłączonym zasilaniu.
4. Do mikrokontrolera należy wgrać oprogramowanie za pomocą zewnętrznego programatora
5. W miejsce 2 złączy 6 pinowych należy umieścić wyświetlacz 7 segmentowy, pamiętając o odpowiednim ustawieniu
6. W złącza mikrokontrolera oraz ekspandera należy umieścić właściwe urządzenia i upewnić się o ich stabilnym umocowaniu
7. W miejsce złącza 26 pinowego IDC należy podłączyć zewnętrzną matrycę led, pamiętając o odpowiednim ustawieniu.
8. Poprawnie zmontowany i oprogramowany układ będzie odgrywał melodię startową oraz będzie wyświetlał ekran powitalny gry „TETRIS”

6. Wnioski z uruchamiania i testowania

6.1 Problemy jakie wystąpiły podczas montażu oraz uruchamiania

Pierwszy problem jaki napotkałem, to brak możliwości zaprogramowania układu za pomocą układu Arduino Leonardo, ze względu na jego niekompatybilność. Konieczne okazało się zamówienie płytki Arduino Uno z wymowanym mikrokontrolerem.

Zastosowanie przerwań w celu multipleksowania wyświetlacza, również okazało się problemem, w przypadku użycia gotowych bibliotek ekspandera zastosowanego w moim projekcie. Wyświetlacz nie pracował w sposób prawidłowy. Konieczne okazało się przełączenie niektórych wyprowadzeń w projekcie, w celu zwolnienia wyprowadzeń dla wyświetlacza i użycie przerwań na pinach znajdujących się bezpośrednio w mikrokontrolerze, a nie w ekspanderze wyprowadzeń.

Kolejnym problemem było wlutowanie niedziałającego przycisku oraz jego późniejsze odlutowanie. Pierwotnie zakładałem, że to brak ciągłości przewodu na pewnym odcinku, jednak finalnie okazało się, że przycisk jest niesprawny i konieczna jest jego wymiana. Niestety nie posiadałem zapasowych przycisków z kompletu, więc zastosowałem przycisk tact-switch z innej serii.

Podczas pierwszego prototypowania na płytce stykowej układ nie działał poprawnie, następowały momenty w których program się zawieszał. Problem udało się rozwiązać stosując dodatkowe kondensatory filtrujące na zasilaniu.

6.2 Testy poprawności działania urządzenia

Zmontowany układ został poddany obszernym testom, w których sprawdzałem reakcje układu na wszystkie możliwe scenariusze. Przetestowana została również logika gry, poprawność zliczania punktów oraz moment zakończenia gry. Wszystkie reakcje są zgodne z założeniami programisty oraz nie wykazują żadnych nieprawidłowości.

7. Wnioski z projektu.

Dzięki wykonaniu tego projektu miałem możliwość zaprojektować oraz wykonać urządzenie elektroniczne oparte na mikrokontrolerze.

Na początku pracy z projektem, bardzo ważnym elementem był odpowiedni dobór elementów. Był on poprzedzony wnikliwą analizą oraz porównaniem wielu rozwiązań. Urządzenia miało być czymś praktycznym a jednocześnie wyróżniającym się na tle innych projektów.

Kolejnym etapem było stworzenie schematu ideowego wraz z schematem blokowym. Był to proces przeniesienia idei na konkretne założenia oraz ich faktyczną realizację. W układzie zostały zastosowane kompaktowe elementy, ze względu na mobilny charakter projektu. W projekcie zostały zastosowane popularne i sprawdzone elementy o bogatej dokumentacji.

Etap montażu elementów został zapoczątkowany projektem ułożenia elementów na płytce uniwersalnej. W następnej kolejności układ został zlutowany i sprawdzony pod kątem poprawności połączeń elektronicznych. Program mikrokontrolera został napisany i przetestowany jeszcze na etapie prototypowania na płytce stykowej. Całość została wgrana na mikrokontroler za pomocą płytki prototypowej Arduino Uno. Dzięki możliwości samodzielnego montażu miałem okazję podszkolić swoje umiejętności lutowania elementów elektronicznych.

Podsumowując, realizacja projektu była zadaniem niezwykle ciekawym i przyjemnym, ale z drugiej strony wymagającym. Najtrudniejszym etapem dla mnie było pisanie oprogramowania oraz szukanie błędów w niedziałającym kodzie. Najprzyjemniejszym etapem zaś było lutowanie i montaż układu. Było to bardzo czasochłonne zajęcie, stąd musiałem zrobić na raty. Finalny efekt bardzo mnie zadowala i będzie zajmować honorowe miejsce w moim domu.

8. Literatura

<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>

https://www.ti.com/lit/ds/symlink/pcf8574.pdf?ts=1613103629765&ref_url=https%253A%252F%252Fwww.google.com%252F

https://www.waveshare.com/wiki/RPi_LED_Matrix

Laboratorium SMIW – przerwania