

Karta projektu zaliczeniowego

Systemy mikroprocesorowe - 2014

Temat projektu: **Gra zręcznościowa dla dwóch osób**

Imię i nazwisko: **Arkadiusz Chorian**

Politechnika Poznańska

kierunek: **AiR**, grupa: **A2**, nr albumu: **109569**

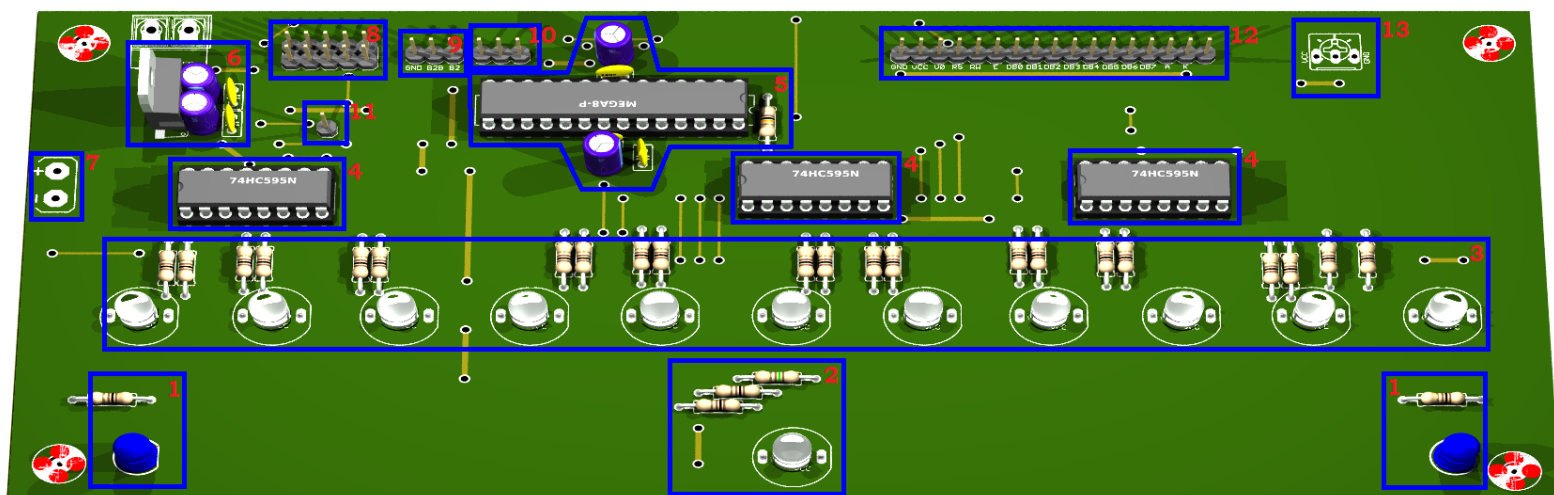
1. Opis projektu

Projekt o nazwie „LED Fight”, to autorska gra zręcznościowa dla dwóch osób. Zarówno pomysł jak i realizacja są rezultatem moich własnych koncepcji. Jednakże podczas realizacji przedsięwzięcia posiłkowałem się ogólnodostępnymi informacjami i rozwiązaniami problemów z wielu dziedzin technicznych.

Zdjęcia układu



Model 3D



1. Diody informujące o zebraniu bonusu relaksu
2. Dioda bonusów
3. Pola walki
4. Rejestry przesuwne 74HC595
5. Mikrokontroler ATmega8A
6. Moduł zasilania ze stabilizatorem 7805
7. Złącze zasilania
8. Złącze programatora ISP
9. Złącze pada gracza 2
10. Złącze pada gracza 1
11. Wyprowadzenie kanału pomiarowego ADC
12. Złącze wyświetlacza LCD
13. Potencjometr kontrastu wyświetlacza LCD

Opis projektu

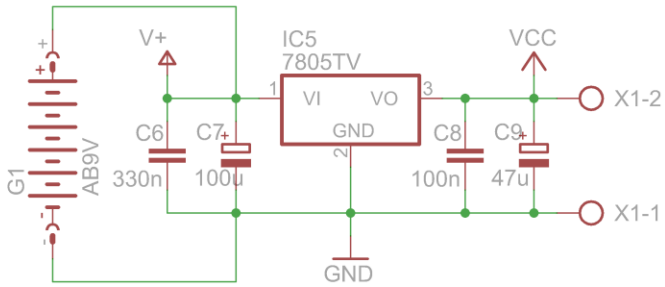
Gra LED Fight pozwala sprawdzić swoją zręczność w potyczce z przeciwnikiem i pomimo swej prostoty zapewnia przyjemną rozrywkę. Zasadę działania gry najlepiej opisać słowem „przepychanka”. Do dyspozycji mamy 11 pól do walki. Gracze ustawiają się naprzeciw sobie. Gdy gra się rozpoczyna, pola walki ustawione są w pozycji startowej. Oznacza to, że gracze posiadają po swojej stronie po tyle samo własnych pól, a środkowe pole jest polem starcia. Przykładowo gracz zielony posiada po swojej stronie 5 zielonych pól, gracz niebieski, po swojej stronie 5 niebieskich pól, a środkowe pole jest koloru turkusowego (wynika z połączenia zielonego z niebieskim). Celem gry jest zdobycie wszystkich pól. Aby to osiągnąć, każdy z graczy ma do dyspozycji swój pad z dwoma przyciskami. Przycisk lewy (zielony) odpowiada za zbieranie bonusów, przycisk prawy (czerwony) odpowiada za walkę o pola. Po starcie gracze powinni jak najszybciej naciskać przyciski walki. Jeśli, któryś z graczy będzie naciskał odpowiednio szybciej niż przeciwnik, zyska pole. Oprócz tego w grze funkcjonuje pole bonusowe. Pole to może przyjąć trzy kolory reprezentujące bonus możliwy do zebrania przez graczy poprzez naciśnięcie przycisku zbierania bonusów na swoim padzie. Pierwszy bonus, koloru czerwonego, jest bonusem odejmującym. W przypadku jego zebrania tracimy natychmiast jedno pole. Drugi bonus, koloru zielonego, to bonus dodający. Analogicznie do poprzedniego, jego zebranie skutkuje uzyskaniem natychmiast jednego pola. Trzeci bonus, koloru niebieskiego, jest tak zwanym bonusem relaksu. Jego zebranie powoduje, że gracz, który go uzyskał może odpocząć od walki przez określoną ilość czasu, podczas gdy przeciwnik nieprzerwanie musi walczyć, by utrzymać pozycję. Informacja o tym, który z graczy zebrał bonus relaksu znajduje się na dedykowanym dla każdego gracza polu, umiejscowionym po jego stronie. Kiedy gracz zbierze ten bonus, pole to zapala się na niebiesko. Gra trwa tak długo, aż jeden z graczy nie uzyska wszystkich pól, włącznie z polem środkowym.

Użyte elementy elektroniczne

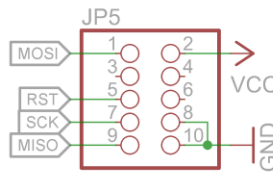
- **Mikrokontroler ATmega8A** – jednostka sterująca
- **Rezystory :**
 - 10 k Ω (x1) – rezystor podciągający pin RST, mikrokontrolera ATmega8 do VCC
 - 100 Ω (x26) – rezystory ograniczające prąd płynący przez diody, dopasowane do koloru niebieskiego i zielonego
 - 150 Ω (x1) – rezystor ograniczający prąd płynący przez diodę koloru czerwonego
- **Kondensatory elektrolityczne :**
 - 10 μ F (x2) – magazynowanie energii
 - 100 μ F (x1) – redukcja tętnień napięcia
 - 47 μ F (x1) – magazynowanie energii
- **Kondensatory ceramiczne/tantalowe :**
 - 100 nF (x3) – filtrowanie zakłóceń
 - 100 nF (x1) – kondensator ustalający napięcie odniesienia dla ADC
 - 330 nF (x1) – filtrowanie zakłóceń
- **Potencjometr 10 k Ω (x1)** – regulacja kontrastu wyświetlacza LCD
- **8-bitowe rejestry przesuwne 74HC595 (x3)** – obsługa większej ilości diod LED
- **Stabilizator napięcia L7805CV** – stabilizacja napięcia wejściowego na 5 V
- **Złącze śrubowe dwupinowe** – podłączenie zasilania
- **Goldpiny** – do połączenia programatora, padów i wyświetlacza LCD
- **Taśma łączeniowa** – podłączenie wyświetlacza LCD
- **Podstawki pod układy scalone**

Schemat układu – część 1

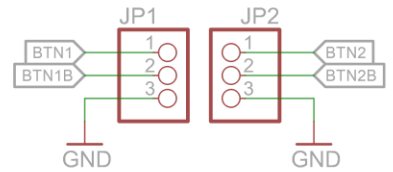
ZASILANIE



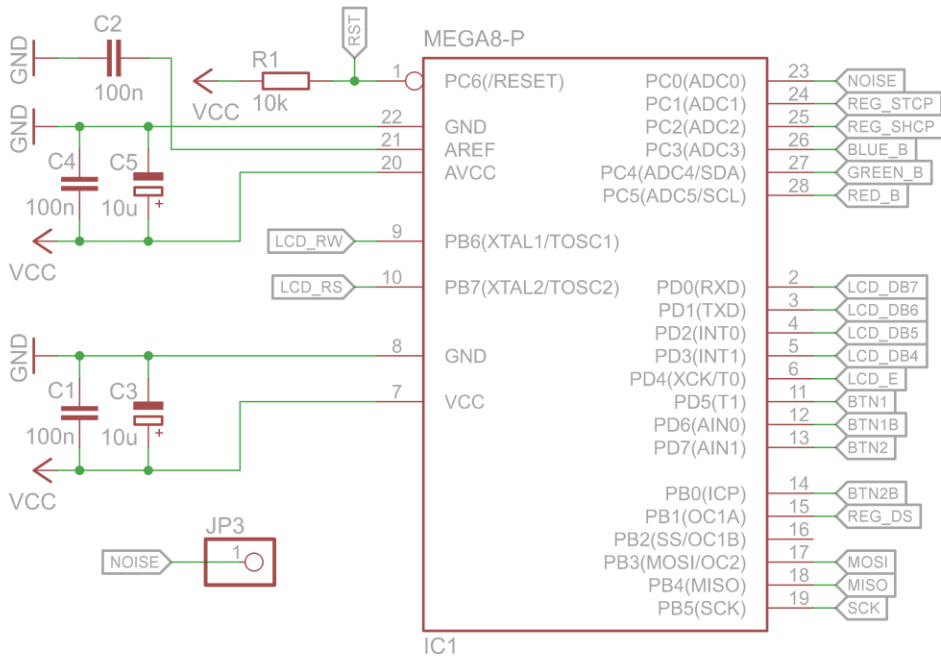
ISP



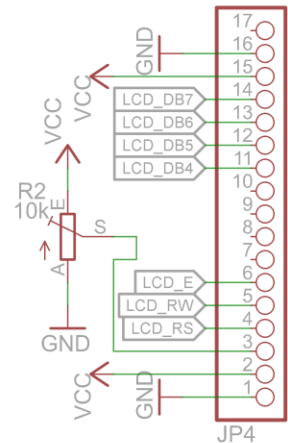
PRZYCISKI



STEROWNIK μ C



LCD



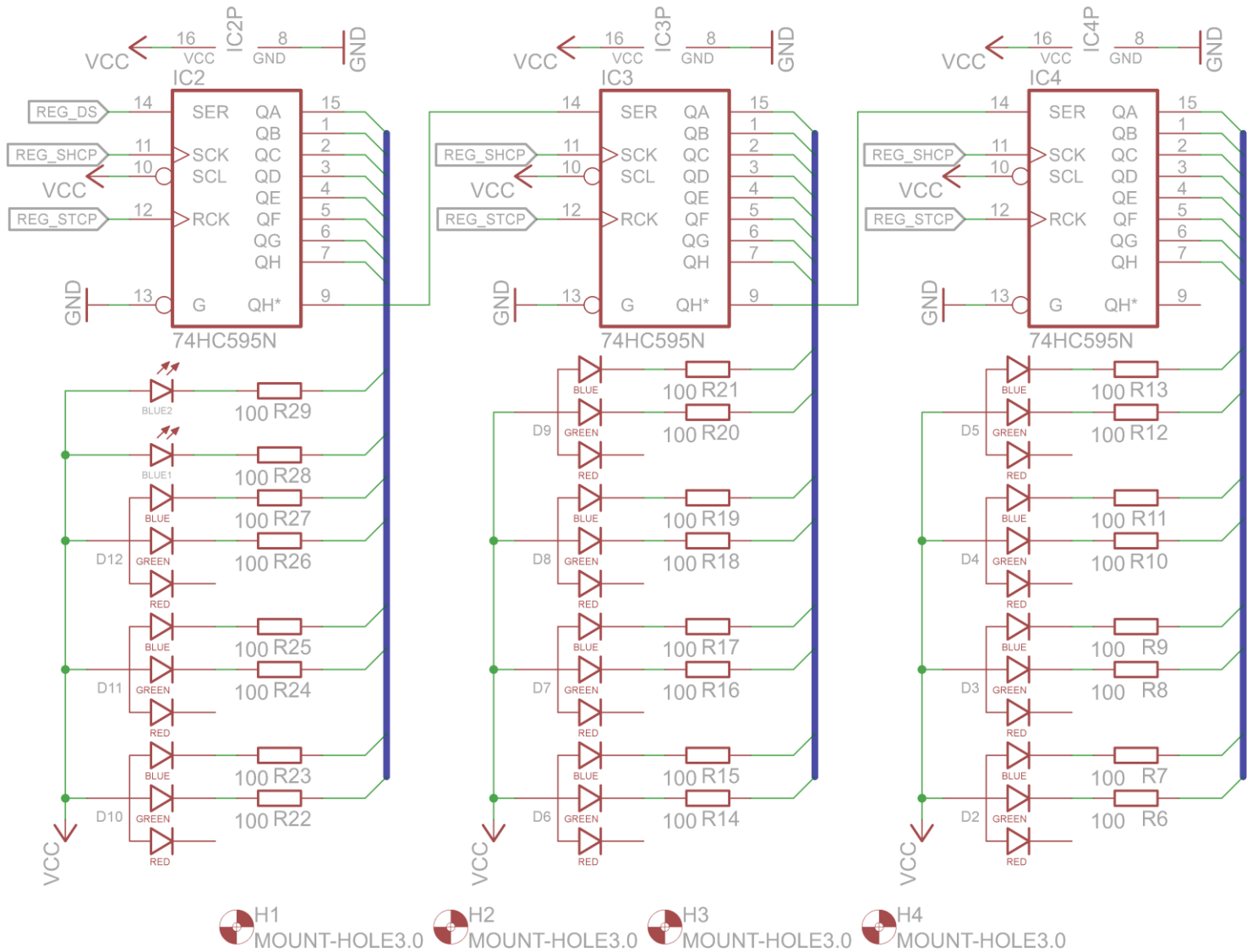
BONUS LED



Sekcje na schemacie części 1

- **Zasilanie** – stabilizator 5V, z kondensatorami poprawiającymi przebieg napięcia oraz z wyprowadzeniami napięcia po stronie wejściowej i wyjściowej
- **ISP** – złącze typu goldpin programatora, w standardzie KANADA
- **Przyciski** – złącza typu goldpin do podłączenia padów
- **LCD** – wyprowadzenie sygnałów z μ C na wyświetlacz LCD ze sterownikiem HD44780 przez złącze typu goldpin
- **Bonus LED** – wyprowadzenie sygnałów z μ C na diodę bonusową
- **Sterownik μ C** - wyprowadzenia sygnałów z μ C, jego zasilanie oraz wyprowadzenie na złącze typu goldpin kanału pomiarowego ADC, celem lepszego zbierania szumów otoczenia

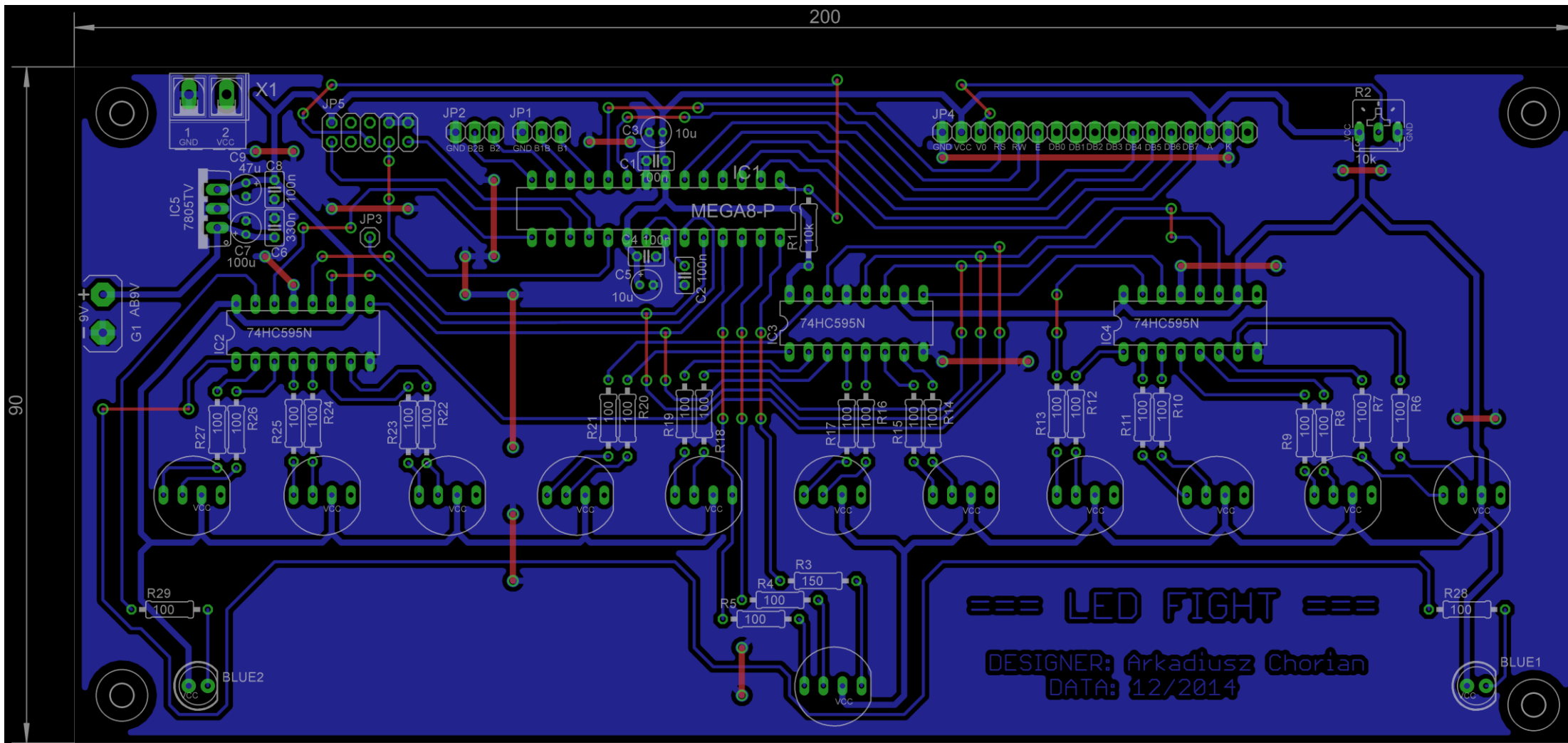
DIODY LED



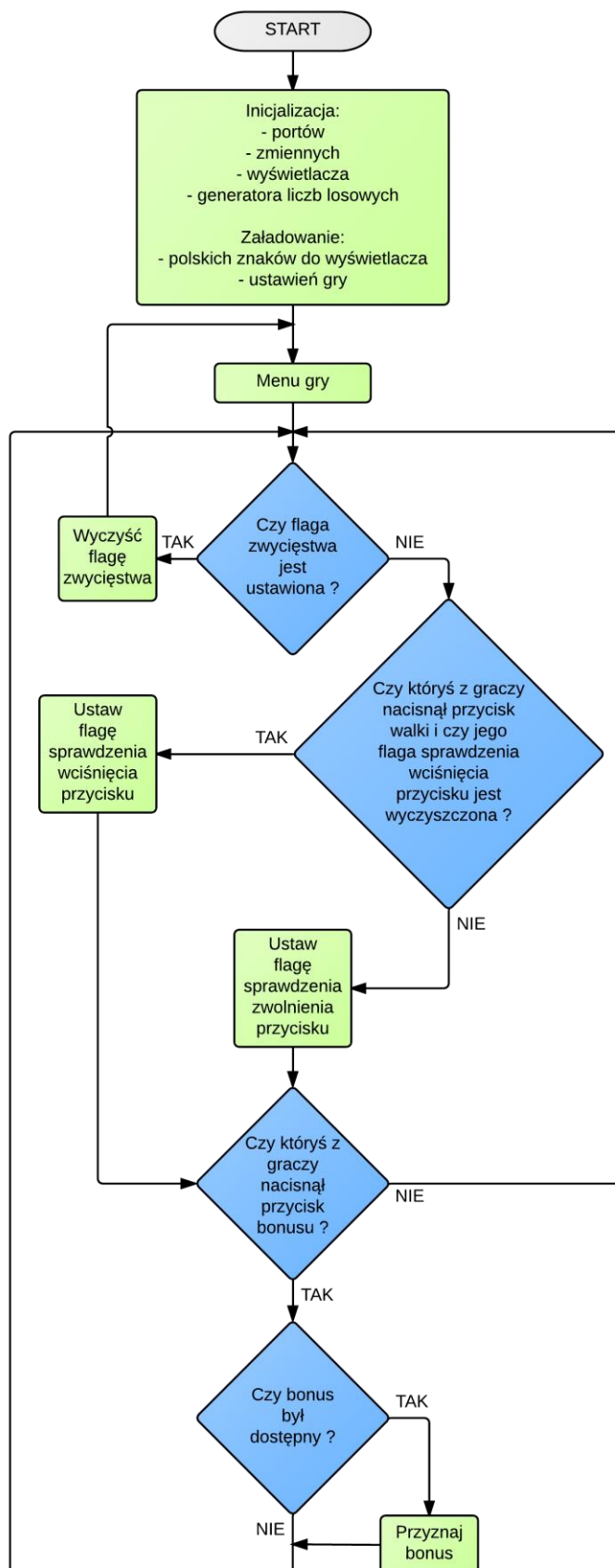
Sekcje na schemacie części 2

- **Diody LED** – zawiera diody LED wraz z rejestrkami przesuwными, które je obsługują oraz 3 mm otwory montażowe

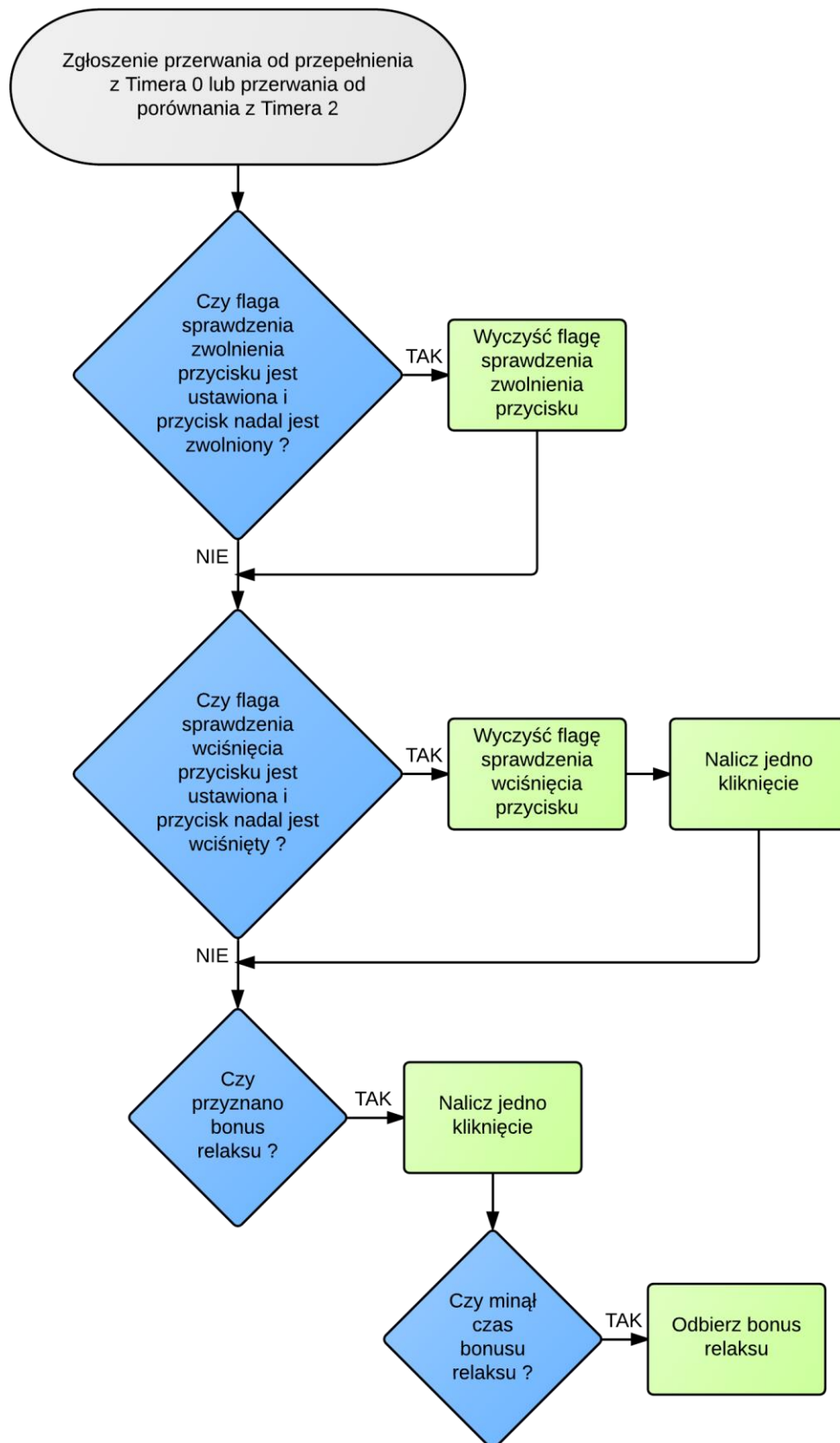
Projekt płytki PCB



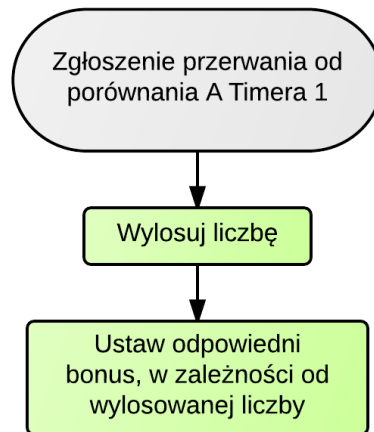
Schemat blokowy – program główny



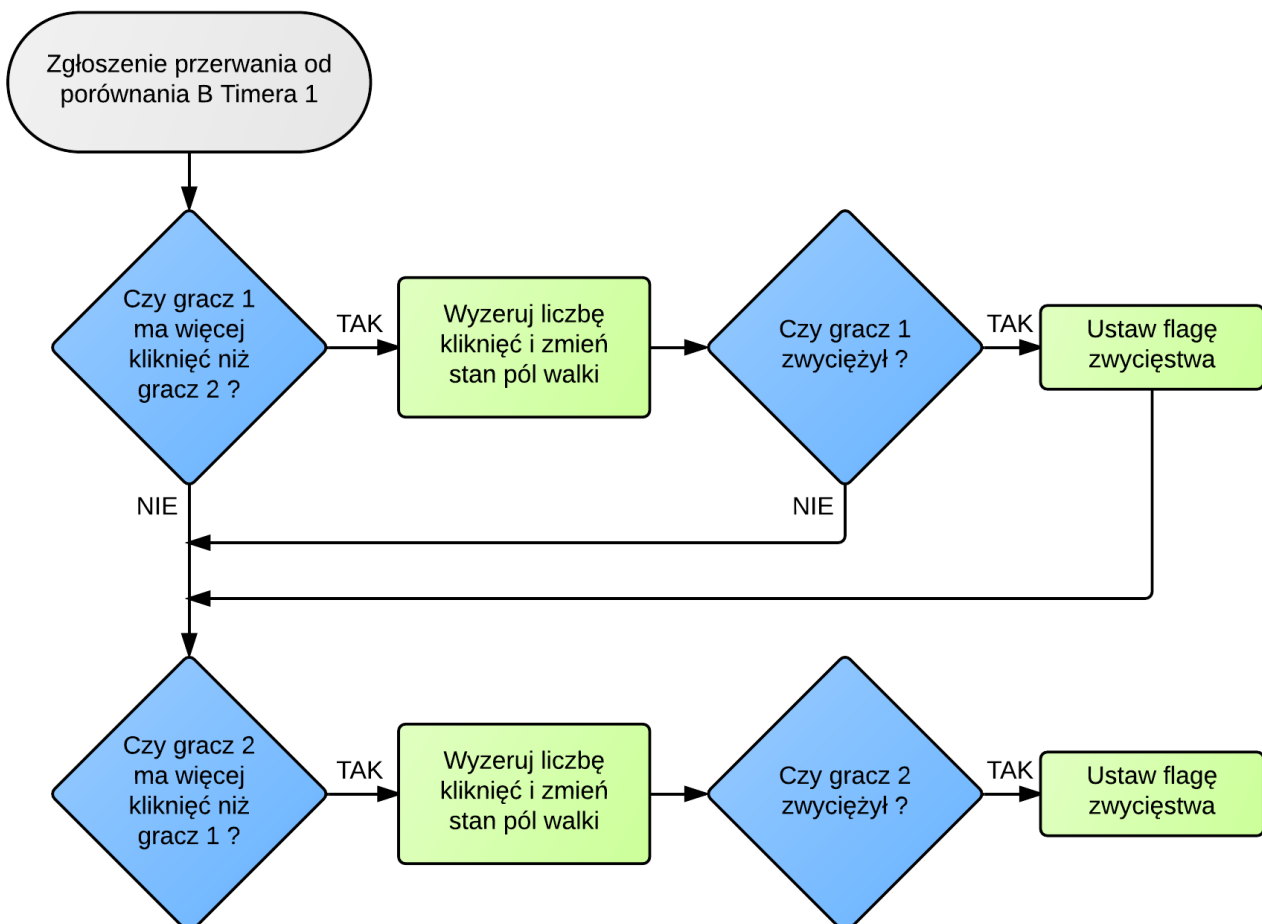
Schemat blokowy – obsługa przycisków



Schemat blokowy – obsługa bonusów



Schemat blokowy – obsługa zmian stanu pola walki



Wykorzystane biblioteki

Nazwa biblioteki	Opis
<avr/io.h>	Standardowa biblioteka kompilatora avr-gcc. Pozwala obsługiwać podstawowe funkcje I/O μ C.
<avr/interrupt.h>	Standardowa biblioteka kompilatora avr-gcc. Pozwala obsługiwać przerwania.
<util/delay.h>	Standardowa biblioteka kompilatora avr-gcc. Pozwala wprowadzać opóźnienia w programie.
<avr/eeprom.h>	Standardowa biblioteka kompilatora avr-gcc. Pozwala obsługiwać pamięć EEPROM μ C.
<avr/pgmspace.h>	Standardowa biblioteka kompilatora avr-gcc. Pozwala obsługiwać pamięć FLASH μ C.
<stdlib.h>	Standardowa biblioteka kompilatora avr-gcc. Pozwala generować liczby losowe.
"lcd44780.h"	Biblioteka stworzona przez Mirosława Kardasia. Pozwala łatwo obsługiwać wyświetlacz LCD.

Najważniejsze funkcje programu

Deklaracja funkcji	Opis
void _delay_ms (int time)	Wprowadza opóźnienie.
inline void shift_bits (uint8_t direction)	Odpowiada za modyfikacje zmiennej określającej stan pola walki.
void change_led_states (uint8_t direction)	Odpowiada za przekazanie stanu pola walki oraz stanu bonusów relaksu graczy na rejestry przesuwne.
void check_last_settings (void)	Odpowiada za sprawdzenie czy w pamięci EEPROM dostępne są poprawne ostatnie ustawienia i ładuje je do pamięci RAM. Jeśli ostatnie ustawienia są niepoprawne (pamięć EEPROM wyczyszczona lub uszkodzona) funkcja ładuje domyślne ustawienia z pamięci FLASH.
void save_current_settings (void)	Zapis obecnych ustawień do pamięci EEPROM.

void game_setup (void)	Menu główne gry.
void buttonX_change_detect (void) void buttonXb_change_detect (void)	Obsługa przycisków obu graczy. Pierwsza funkcja współpracuje z przerwaniem Timerów, pozwalając obsługiwać drgania styków. Druga funkcja odpowiada za reakcję na zebranie bonusu.
void program_polish_characters (void)	Zaprogramowanie polskich znaków we wyświetlaczu LCD HD44780.
void init_XXX (void)	Funkcje inicjalizujące moduły sprzętowe, porty i zmienne.

4. Wykorzystane narzędzia projektowe

Wykorzystane oprogramowanie

- Atmel Studio 6.2 – oficjalne IDE przygotowane do programowania mikrokontrolerów firmy ATmel. Środowisko jest nakładką na IDE Microsoft Visual Studio.
- AvrDude – niewielki program pozwalający wgrać wsad do mikrokontrolera. Został przeze mnie odpowiednio zintegrowany ze środowiskiem Atmel Studio.
- Eagle 7.2 – środowisko do tworzenia schematów elektronicznych oraz projektowania płytek PCB. Dodatkowo wykorzystano wtyczkę Eagle 3d pozwalającą generować model 3D płytki PCB w formacie *.pov
- Pov-Ray 3.7 – program pozwalający renderować obrazy z plików o rozszerzeniu *.pov. Został wykorzystany do stworzenia modelu 3D płytki PCB.
- MkAvrCalculator 1.1.0 b68 – program autorstwa Mirosława Kardasia. Został wykorzystany do ustawienia Fuse Bitów mikrokontrolera oraz do obliczeń związanych z Timerami.
- Lucidchart online – internetowa wersja oprogramowania do tworzenia schematów blokowych
- Microsoft Word 2013 – edytor dokumentów. Wykorzystany do stworzenia dokumentacji.

Programator

- Do zaprogramowania układu użyto programatora USBasp V2.0



5. Weryfikacja poprawności działania układu

Ze względu na rozrywkowo-towarzyskie przeznaczenie układu określenie jakości jego działania ma charakter nieścisły. Jedyną możliwością przetestowania układu były „testy terenowe” i zbadanie ludzkiej, subiektywnej opinii. Na zmontowanym układzie odbyło się kilkadziesiąt rozgrywek różnych osób. Wszystkie starcia przebiegły bez kłopotów ze strony urządzenia. Zdaniem części uczestników testu gra jest przyjemna, wciągająca oraz pomimo swej prostoty, bardzo grywalna. Do momentu oddania projektu nie zauważono żadnych problemów związanych z układem.

Nawigacja po menu

Po menu poruszamy się za pomocą obu padów, według poniższej tabeli :

Pad gracza	Przycisk	Miejsce	Akcja
1	PRAWY	MENU GŁÓWNE	WEJŚCIE W PODMENU/WYBÓR
1	LEWY	MENU GŁÓWNE	POZYCJA STARTOWA (NOWA GRA)
2	PRAWY	MENU GŁÓWNE	NASTĘPNA POZYCJA
2	LEWY	MENU GŁÓWNE	POPZEDNIA POZYCJA
1	PRAWY	USTAWIENIA	ZATWIERDZENIE USTAWIENIA I PRZEJSCIE DO NASTĘPNEGO
1	LEWY	USTAWIENIA	ZAPIS USTAWIEŃ I WYJŚCIE DO GŁÓWNEGO MENU
2	PRAWY	USTAWIENIA	ZWIĘKSZENIE WARTOŚCI
2	LEWY	USTAWIENIA	ZMNIĘJSZENIE WARTOŚCI

Ustawienia

➤ Liczba pól



Dostępne opcje [3, 5, 7, 9, 11]

Określa ile pól wykorzystywanych jest podczas rozgrywki. Maksymalna liczba pól do wykorzystania to 11. Przy wybraniu mniejszej wartości pomijane pola pozostają nieaktywne.

Domyślna wartość : 11.

➤ Czułość padów



Dostępne opcje [1, 2, 3, 4]

Określa jak duża musi być różnica pomiędzy ilością przyciśnień klawisza walki, aby zyskać jedno pole. Przykładowo, jeżeli parametr ten jest równy 1, aby zdobyć jedno pole, gracz musi nacisnąć przycisk walki tylko raz, zakładając, że w tym czasie przeciwnik nie naciska. Kiedy parametr ten wynosi 4, aby zdobyć pole, należy nacisnąć 4 razy. Każde naciśnięcie przycisku przez przeciwnika w czasie kompletowania 4 przyciśnień redukuje naliczone przyciśnięcia o 1. W ogólności uzyskanie jednego pola warunkowane jest zależnością:

Przyciśnięcia gracza X > Przyciśnięcia gracza Y + Czułość padów \Rightarrow X zyskuje pole

Domyślna wartość : 2

➤ Częstość bonusów

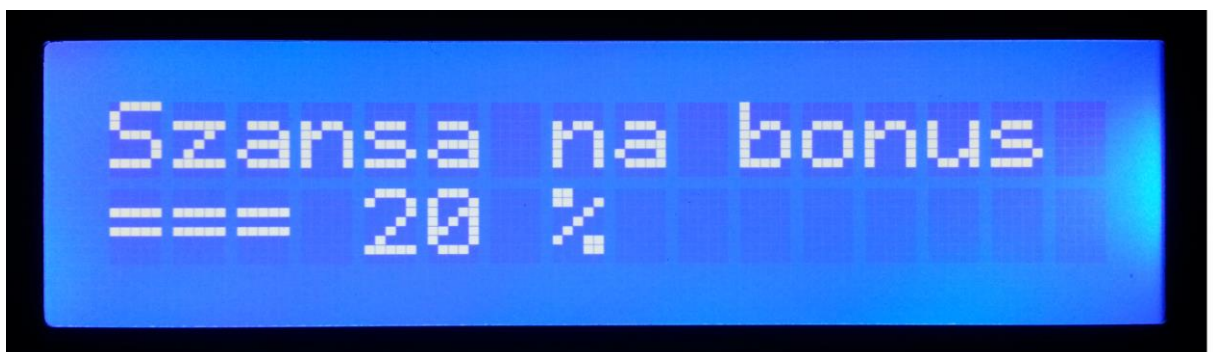


Dostępne opcje [od 100ms do 2000ms, co 100ms]

Określa, co jaki czas podejmowana jest próba losowa, mogąca potencjalnie wygenerować bonus. O wystąpieniu lub niewystąpieniu bonusu decyduje parametr [Szansa na bonus].

Domyślna wartość : 2000ms.

➤ Szansa na bonus

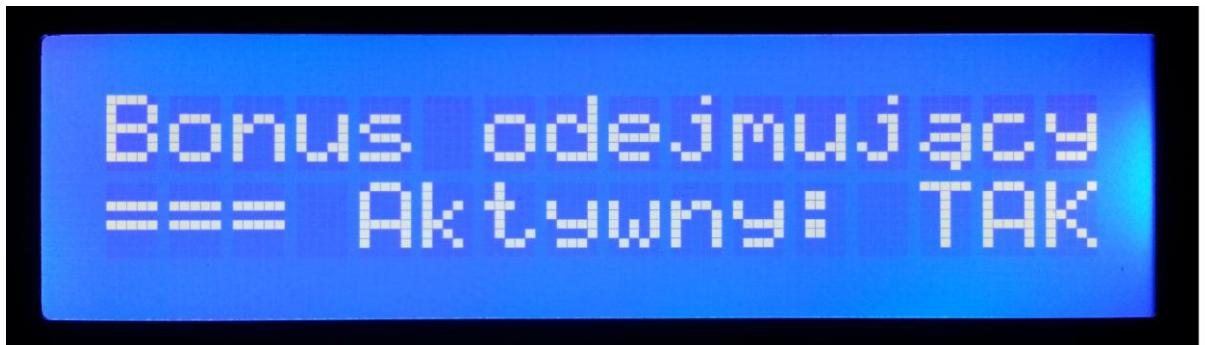


Dostępne opcje [10%, 11%, 12%, 14%, 16%, 20%, 25%, 33%]

Określa jakie jest prawdopodobieństwo wystąpienia pojedynczego bonusu, podczas podejmowania próby losowej. O częstości podejmowania prób losowych decyduje parametr [Częstość bonusów].

Domyślna wartość : 20%.

➤ Bonus odejmujący

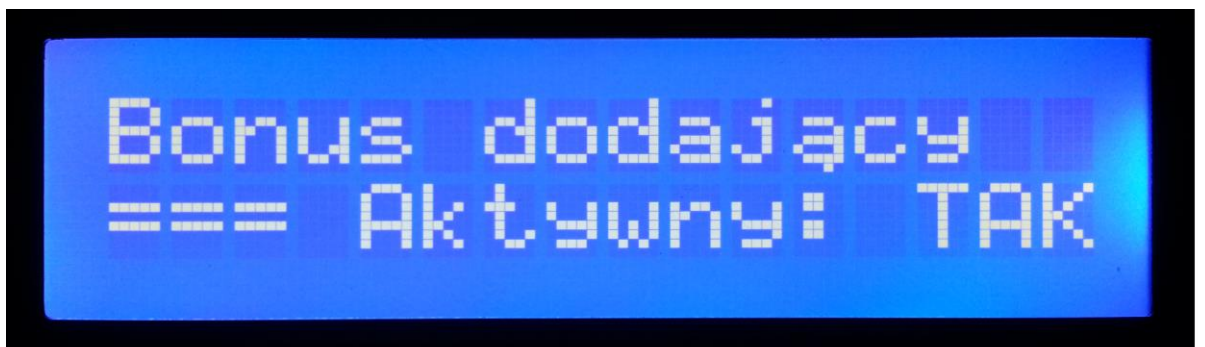


Dostępne opcje [TAK, NIE]

Określa czy podczas rozgrywki bonus odejmujący ma być aktywny. Ustawienie opcji NIE powoduje, że bonus nie będzie generowany.

Domyślna wartość : TAK

➤ Bonus dodający

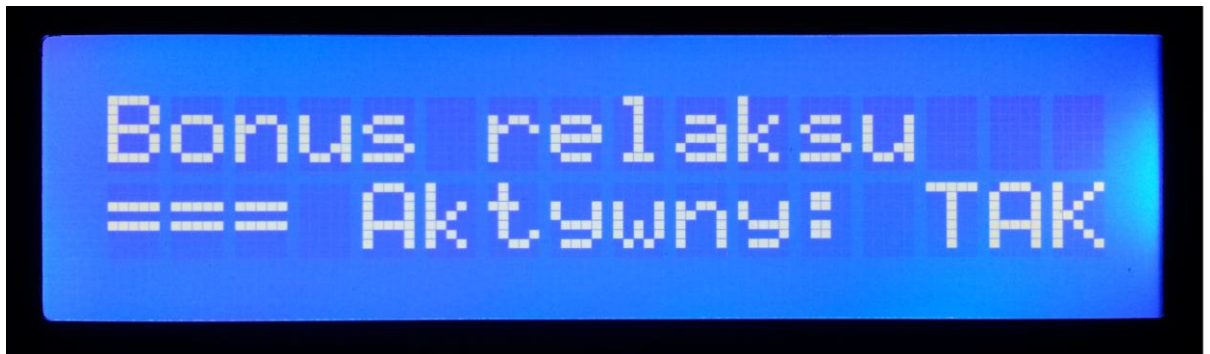


Dostępne opcje [TAK, NIE]

Określa czy podczas rozgrywki bonus dodający ma być aktywny. Ustawienie opcji NIE powoduje, że bonus nie będzie generowany.

Domyślna wartość : TAK

➤ Bonus relaksu

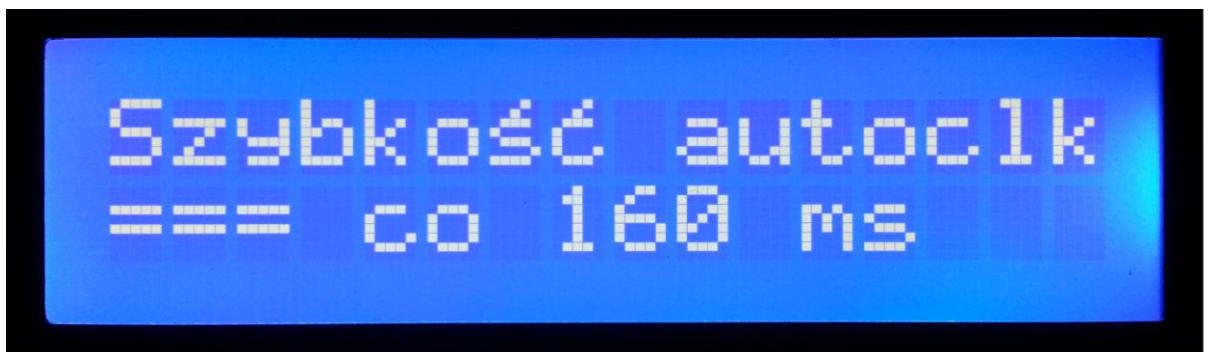


Dostępne opcje [TAK, NIE]

Określa czy podczas rozgrywki bonus relaksu ma być aktywny. Ustawienie opcji NIE powoduje, że bonus nie będzie generowany.

Domyślna wartość : TAK

➤ Szybkość autoclick

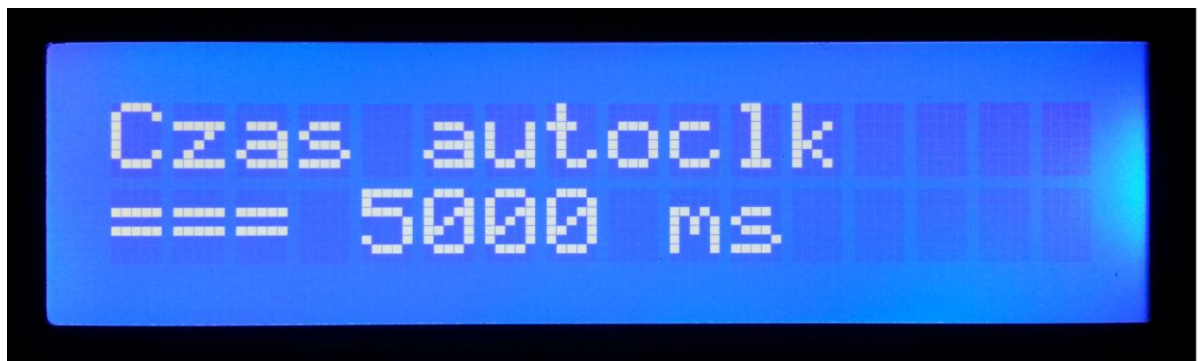


Dostępne opcje [od 100ms do 1000ms, co 20ms]

Określa jaka będzie szybkość sztucznego naciskania, przy zebranych bonusie relaksu. Przykładowo wartość 100ms oznacza, że kiedy gracz zbierze bonus relaksu, jego przeciwnik będzie musiał naciskać przycisk walki co najmniej z szybkością jednego naciśnięcia co 100ms, aby nie stracić pola.

Domyślna wartość : 160ms

➤ Czas autoclick



Dostępne opcje [od 1000ms do 20000ms, co 100ms]

Określa jak długo będzie trwało sztuczne naciskanie, przy zebranych bonusie relaksu. Przykładowo wartość 1000ms oznacza, że kiedy gracz zbierze bonus relaksu, nie będzie musiał sam naciskać przycisku walki przez 1000ms.

Domyślna wartość : 5000ms

➤ Automatyczne wygaszanie bonusu odejmującego



Dostępne opcje [TAK, NIE]

Określa czy bonus odejmujący, będzie automatycznie wyłączany przy następnej próbie losowej, która nie wygeneruje innego bonusu.

Domyślna wartość : TAK

Parametry elektryczne

Napięcie zasilania [V]			Pobór prądu [mA]		
Minimalne	Typowe	Maksymalne	Minimalny	Typowy	Maksymalny
7	9	20	75	83	91

7. Literatura

1. Dokumentacja mikrokontrolera ATmega8A

http://www.atmel.com/images/atmel-8159-8-bit-avr-microcontroller-atmega8a_datasheet.pdf

2. Dokumentacja rejestru przesuwneego 74HC595

http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf

3. Dokumentacja stabilizatora L7805CV

<http://datasheet.octopart.com/L7805CV-STMicroelectronics-datasheet-7264666.pdf>

4. Dokumentacja diod LED RGB

<https://www.sparkfun.com/datasheets/Components/YSL-R596CR3G4B5C-C10.pdf>

5. Dokumentacja wyświetlacza LCD WC1602A (sterownik HD44780)

<http://osworld.pl/wp-content/uploads/WC1602A-STBLWHTC-06.pdf>

6. „Mikrokontrolery AVR. Język C. Podstawy programowania”, Mirosław Kardaś, wyd. Atmel, 2011

7. Blog Mirosława Kardasia dotyczący mikrokontrolerów - <http://mirekk36.blogspot.com/>

8. Blog dotyczący mikrokontrolerów - <http://mikrokontrolery.blogspot.com/>

9. Kurs obsługi programu Eagle, Mirosława Kardasia

9.1. Część 1 - <https://www.youtube.com/watch?v=5CF0Q81nGzY>

9.2. Część 2 - <https://www.youtube.com/watch?v=tTPnptX5vvM>

9.3. Część 3 - <https://www.youtube.com/watch?v=CfZ4knllq2s>

9.4. Część 4 - <https://www.youtube.com/watch?v=873BEoIHsa8>

9.5. Część 5 - <https://www.youtube.com/watch?v=tHIRrkGH0GM>

9.6. Część 6 - <https://www.youtube.com/watch?v=0TkAmNeBjXU>