

**POLITECHNIKA ŚLĄSKA  
WYDZIAŁ ELEKTRYCZNY**

Kierunek: Mechatronika

Specjalność:  
Zastosowanie mechatroniki w  
inżynierii elektrycznej

Rodzaj studiów:  
stacjonarne I stopnia

**Projekt inżynierski**

**Paweł Pawlaszczyk**

**Koncepcja, projekt oraz wykonanie  
układu do badania refleksu  
snajpera strzelającego  
z broni typu ASG.**

**Project, design and construction  
of a system to study reflex of  
an sniper shooting using  
ASG type weapon.**

Kierujący pracą:

Dr inż. Jarosław Domin

Recenzent:

Dr Inż. Zbigniew Pilch

**Gliwice, Marzec, 2018r.**

## Spis treści

1.	Wstęp .....	4
2.	Cel i założenia .....	6
3.	Opis teoretyczny projektu.....	8
3.1.	Wybór języka oprogramowania.....	8
3.1.1.	iTextSharp – Biblioteka C# .....	10
3.1.2.	Media – Biblioteka C#.....	10
3.1.3.	Data.SqlClient – Biblioteka C# .....	11
3.1.4.	IO – Biblioteka C#.....	11
3.2	MySQL .....	11
3.3	Arduino.....	12
3.3.1.	Układ MPU6050 – Biblioteka Arduino.....	14
3.4.	Środowisko programistyczne.....	14
3.4.1.	Microsoft Visual Studio .....	14
3.5.	Środowisko CAD .....	15
3.5.1.	Autodesk Inventor Proffesional.....	16
3.5.2.	FEM – Autodek Inventor Proffesional .....	16
3.6.	Obliczanie sprężyny śrubowej naciskowej .....	21
3.7.	Akcelerometr .....	23
4.	Wdrożenie.....	24
4.1.	Układ mechaniczny.....	25
4.1.1.	Dobór sprężyny i materiału .....	26
4.2.	Układ elektroniczny .....	27
4.2.1.	Widok programu arduino .....	28
4.2.2.	Połączenie układu elektronicznego z mechanicznym .....	30
4.3.	Interfejs programu.....	30
4.3.1.	Komunikacja z bazą danych .....	31
4.3.2.	Zabezpieczenie programu.....	35
4.3.3.	Interfejs trybu refleks .....	37

5.	Testy aplikacji, BHP, przyszły rozwój .....	41
5.1.	Testy aplikacji.....	42
5.2.	BHP.....	43
5.3.	Rozwój aplikacji w przyszłości .....	44
6.	Podsumowanie projektu .....	45
7.	Bibliografia.....	46

# 1. Wstęp

Historia broni palnej rozpoczyna się w IX wieku w Chinach. Początek tej historii nadało odkrycie czarnego prochu, jako materiału wybuchowego. Początkowo zaczęto go używać jako tzw. paliwo do fajerwerków i do XIX wieku był jedyną znaną i stosowaną mieszanką wybuchową. Następstwem tego odkrycia było skonstruowanie pierwszej broni palnej (Chiny, XIII w.). Postęp technologiczny, który dokonał się na przełomie wieków, nie tylko w znaczny sposób ją zrewolucjonizował, ale również wpłynął na spopularyzowanie jej zastosowania. Obecnie jest używana nie tylko do celów militarnych np. w wojsku, ale również do celów czysto rozrywkowych. Skupiając się nieco bardziej na celach rozrywkowych nie da się nie zauważyć, że w dzisiejszych czasach ten rodzaj korzystania z broni jest bardzo popularny. Posiadanie broni przez „zwykłego” obywatela w wielu krajach, np. w Polsce, jest zakazane. Ponadto strzelnice, które świadczą tego typu usługi są drogie. Dlatego też prawdziwi pasjonaci znaleźli na to sposób i stworzyli Air Soft Gun (ASG), czyli replikę broni palnej wykorzystywaną w celach hobbystycznych. [1]

W wyniku tego, w projekcie inżynierskim, zostały podjęte działania mające na celu opracowanie strzelnicy służącej celom rozrywkowym. Projekt jest dostosowany do użytku broni typu ASG, a jego celem jest określenie refleksu snajpera. Ten wykonując strzał, z replik broni o prędkości wylotowej mieszczącej się w granicach od 121 m/s do 152 m/s, zostanie poinformowany o trafieniu poprzez diodę umieszczoną nad tarczą.

Oprogramowanie dedykowane jest dla systemu Windows, najbardziej popularnego w dzisiejszych czasach oprogramowania przeznaczonego dla użytku osobistego. Podczas pisania głównej aplikacji wykorzystano język C#, gdzie użyto szablonu projektu aplikacji formularzy systemu Windows oprogramowania Microsoft Visual Studio. Użyte szablony formularzy aplikacji połączone są ze sobą programowo, a każdy z poszczególnych formularzy wypełnia przypisane sobie zadanie, ważne dla całego działania programu. Oprogramowanie również odpowiada za obliczenia związane z otrzymanymi wynikami określającymi refleks strzelca oraz za generowanie gotowego pliku „PDF” do odczytu wspomnianych wyników.

Poniżej przedstawiony został proces tworzenia projektu. Na początku opisane są cele i założenia części elektronicznej, mechanicznej i programowej projektu. Następnie podjęto tematy: technologii wykorzystanej podczas tworzenia oprogramowa-

nia, procesu połączenia z bazą danych, opisu dotyczącego oprogramowania, które zostało wykorzystane do zaprojektowania części mechanicznej projektu oraz temat opisu elementów wykorzystanych do zbudowania układu elektronicznego. W dalszej części pracy znajduje się przedstawienie narzędzi, jakie zostały wykorzystane w tym projekcie. Ma ono na celu ukazanie interfejsu oraz sposobu implementacji funkcji. Kolejną częścią pracy jest opis testów przeprowadzanych przy użyciu aplikacji, które wykonywane były w warunkach rzeczywistych i zawierały wszystkie możliwe warianty dotyczące wyniku określającego refleks snajpera z broni typu ASG. Praca zawiera również zasady BHP dotyczące korzystania z broni, mające na celu zapewnić bezpieczeństwo wszystkim użytkownikom. Następny punkt poświęcony jest tematyce rozwoju aplikacji w przyszłości. W zakończeniu pracy znajdują się wnioski wraz z podsumowaniem pracy oraz bibliografia.

## 2. Cel i założenia

Celem pracy jest zaprojektowanie oraz opracowanie układu badającego refleks snajpera oddającego strzał z broni typu ASG. Pracę podzielono na trzy etapy: pierwszy etap pracy związany jest z doбором podzespołów elektronicznych, których zadaniem jest stworzenie układu elektronicznego współpracującego z oprogramowaniem. Następnym tego procesu jest rejestrowanie okresu czasu pomiędzy zapaleniem się diody a trafieniem pocisku w tarczę. Dla pierwszego etapu pracy przyjęto następujące założenia:

- Wykorzystanie czujnika akcelerometrycznego (przyśpieszenia), o stopniu czułości nie mniejszym niż 2G oraz 16 bitowej rozdzielczości przetwornika, do pomiaru chwili uderzenia pocisku w tarczę.
- Zastosowanie mikrokontrolera AVR atmega328 znajdującego się na gotowej płytce Arduino Uno.
- Zastosowanie diody jako lampki sygnalizującej, widocznej wyraźnie z 10 metrów.
- Zastosowanie komputera osobistego jako zasilania dla płytki Arduino Uno z mikrokontrolerem.
- Wykorzystanie komputera osobistego do obsługi aplikacji.

Drugi etap dotyczy części mechanicznej i polega na opracowaniu rysunku technicznego ukazującego wygląd układu mechanicznego, sposób zamocowania jego poszczególnych części oraz materiałów z jakich jest wykonany.

- Dobór materiału do wykonania tarczy, o maksymalnej wytrzymałości na prędkość wylotową 152 m/s dla masy kulki 0,25 g oraz masie nie większej niż 200 g. Zakłada to wprowadzenie w ruch tarczy połączonej ze sprężyną przy najmniejszej prędkości wylotowej 121 m/s i masie 0,25 g pocisku.
- Zabezpieczenie diody materiałem zapewniającym izolację od nietrafiionych w tarczę pocisków wystrzelonych z broni palnej.
- Opracowanie rysunku technicznego profilu o przekroju kwadratowym i zlecenie wykonania tego profilu, będącego kanałem kablowym przewodów elektrycz-

nych prowadzących od płytki Arduino do czujnika umieszczonego pod tarczą i diody znajdującej się nad tarczą.

- Wykorzystanie profilu o przekroju kwadratowym do umieszczenia tarczy połączonej z sprężyną na wysokości nie mniejszej niż 20cm.
- Mocowanie śrubowe profilu o przekroju kwadratowym do płyty wiórowej.
- Dobór sprężyny o sztywności nie mniejszej niż 2,0 N/mm i nie większej niż 4,00 N/mm oraz czasie powrotu do pozycji pierwotnej nie większym niż 2,5 sekundy.

Ostatnia część dotyczy zaprojektowania i zaprogramowania aplikacji. Zadaniem wspomnianej aplikacji jest przesyłanie danych z czujnika umieszczonego pod tarczą do mikrokontrolera. Następnie te informacje są przysyłane na komputer osobisty, gdzie ma miejsce ich programowa interpretacja. Ważnym elementem projektu jest baza danych, w której dokonuje się magazynowanie wyników pomiaru refleksu strzelca. W części trzeciej zawarte są następujące założenia:

- Nieuwzględnianie przez aplikację, przy obliczaniu refleksu strzelca, strzałów nietrafionych w tarczę.
- Wykorzystanie komunikacji szeregowej mikrokontrolera z komputerem (UART) do odbierania danych akcelerometru.
- Sygnalizowanie przez diodę, umieszczoną nad tarczą, rozpoczęcia rejestracji pomiaru czasu pomiędzy oddaniem strzału a momentem uderzenia pocisku w tarczę, w odstępach czasowych od 2,5 do 4,5 sekundy.
- Obliczanie średniej wartości czasowej na podstawie wszystkich wyników uzyskanych przez jednego strzelca oraz wyświetlanie ich w formacie .pdf wraz z informacjami takimi jak: typ broni, maksymalna oraz minimalna wartość czasowa strzału, nazwa strzelca, data uzyskania pomiaru.
- Zapisywanie lokalne bazy danych w systemie MySQL.
- Możliwość dodawania, usuwania i modyfikacji broni wykorzystywanych do pomiaru refleksu strzelca.
- Wykorzystanie zabezpieczania blokującego dostęp do modyfikacji bazy danych przez nieuprawnionych użytkowników podczas korzystania z aplikacji.
- Wykorzystanie w projekcie inżynierskim jednej tarczy do odczytu chwili uderzenia pocisku.

### 3. Opis teoretyczny projektu

Do sporządzenia poprawnie działającego oprogramowania autor projektu musiał dobrać właściwą technologię dopasowaną do każdego z zadań przewidzianych w projekcie. Każdy element pracy inżynierskiej musi być kompatybilny z każdym kolejnym komponentem i nie może powodować konfliktów czy zagrożeń podczas użytkowania. Na tej podstawie przy doborze odpowiedniej technologii warunkiem było połączenie jej z wszystkimi możliwymi rozwiązaniami co było konieczne np. przy połączeniu języka programowania C# z bazą danych MySQL.

#### 3.1. Wybór języka oprogramowania

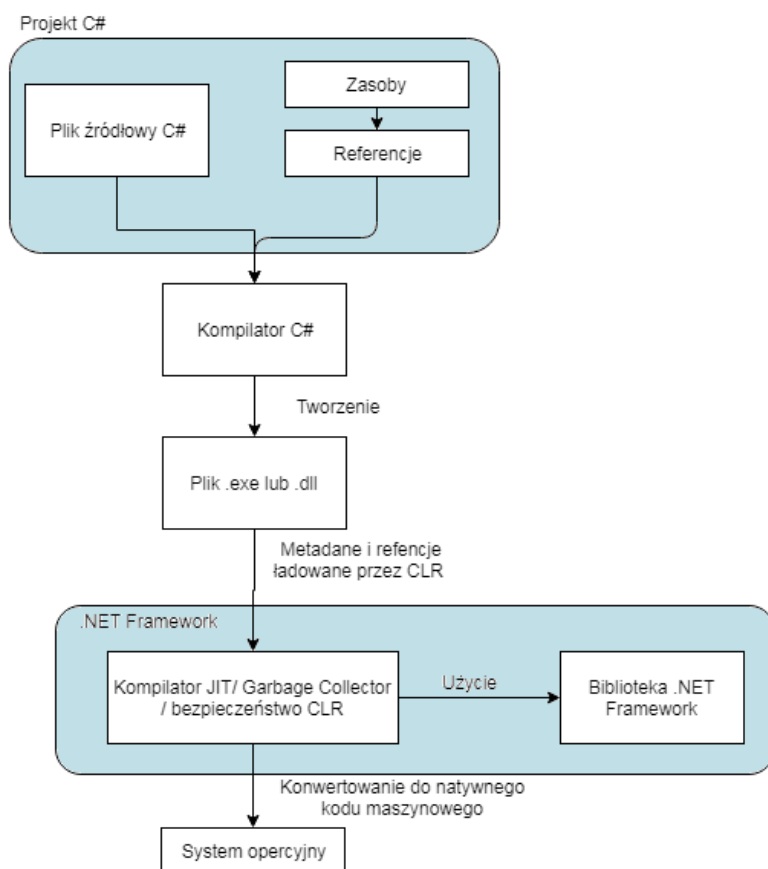
Podczas wyboru języka programowania odpowiedniego do opracowania projektu wzięto pod uwagę umiejętność posługiwania się językiem przez autora pracy oraz aktualność, łatwość użytkowania języka, a także brak potrzeby modernizacji interfejsu i kodu aplikacji przy modyfikacji innych elementów kodu. Ważnym elementem doboru stało się kryterium pojemności pamięciowej oraz kryterium ilości wykorzystywanych zasobów sprzętowych komputera przez język programowania. Zaprojektowane oprogramowanie przeznaczone jest do użytku przez system Windows, gdyż jest on aktualnie najbardziej rozpowszechnionym i najczęściej używanym oprogramowaniem systemowym. Stąd też wybór języka C#, ponieważ stawia on czoła wszystkim wymaganiom systemowym ww. oprogramowania. Język C# jest ciągle rozwijany przez twórców, którzy cały czas dokonują zmian i udoskonaleń (najnowsza wersja z roku 2017), co nadaje mu miano współczesnego obiektowego języka programowania, który umożliwia wykorzystywanie obiektowych metod programowania. Przy pisaniu kodu używając języka C# można skorzystać z pomocy gotowych bibliotek .NET, co sprawia, że tworzenie kodu nie jest problematyczne. Ponadto modyfikacja poszczególnych fragmentów kodu nie sprawia nieprzewidzianych problemów w całym kodzie, ponieważ jest umieszczony pod elementami interfejsu tj. pole tekstowe. Język ten ma również wszechstronne zastosowanie. Poprzez wbudowane CLR (Common Language Runtime) czyli wspólne środowisko uruchomieniowe, mamy możliwość łączenia się z różnymi elementami systemu operacyjnego. Celem ww. środowiska jest odczyt poleceń znajdujących się w CIL (Common Intermediate Language), który jest wspólnym językiem przejściowym. Ostatnim zadaniem języka CLR jest przekształcenie odczytu w kod maszynowy i wykonanie go przez procesor. JIT (Just in time compi-



lation) to kompilator dzięki któremu kompilacja kodu odbywa się w czasie działania programu. Dzięki takiemu rozwiązaniu możemy wyróżnić następujące zalety kodu: [2]

- Kod komputerowy, wspomagany przez CLI służy najlepszym rozwiązaniom. W momencie uruchomienia aplikacji na komputerze, z wbudowanym procesorem 64-bitowym, CLI umożliwia osiągnięcie największych efektów jednocześnie przy najmniejszym wkładzie procesora.
- Możliwość uruchomienia na jakiegokolwiek platformie CLR dla dowolnego kodu programu napisanego językiem wspieranym przez CLI.

Powyższe procesy realizacji programu bazującego na języku C# udowadniają, że spełnia on wszystkie niezbędne kryteria niezbędne do prawidłowego funkcjonowania programu. Ponadto nieprawidłowości, które nie mają miejsca w języku C#, a jakie występowały w pozostałych językach programowania np. wycieki pamięci w programie występujące w języku C++, zaważyły na wyborze tego języka do opracowania głównego oprogramowania projektu.

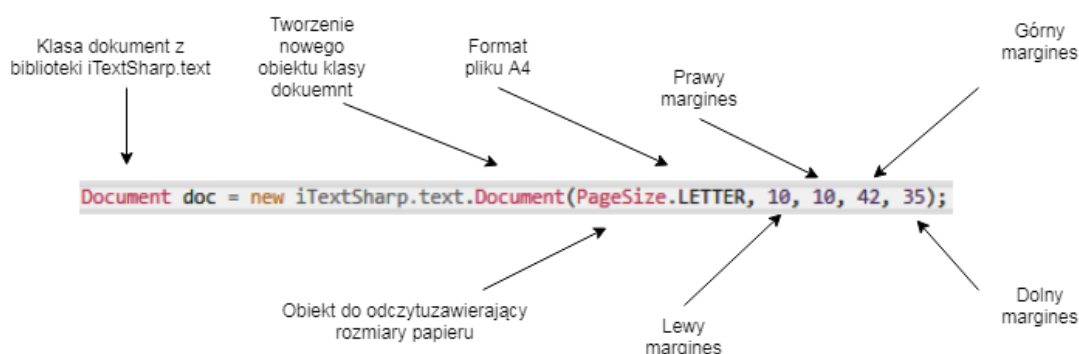


Rys. 3.1. Algorytm działania języka C# ukazujący w sposób graficzny jego efektywność. [3]

Podczas pisania oprogramowania użyto gotowych bibliotek .NET, które ułatwiały realizację celów, jakie wyznaczył autor projektu w punkcie drugim. Najważniejsze z tych bibliotek przedstawiono poniżej.

### 3.1.1. iTextSharp - biblioteka C#

„iTextSharp” jest tzw. biblioteką „open source”, czyli biblioteką wolnego oprogramowania platformy .Net. Używana jest między innymi właśnie w języku C# (również w języku JAVA) i umożliwia obsługiwanie, przeglądanie, dostosowywanie, a także tworzenie dokumentów w formacie PDF (Portable Document Format) inaczej znanym jako format przenośnego dokumentu. Biblioteka ta głównie umożliwia tworzenie plików w wyżej wspomnianym formacie wykorzystując w sposób dynamiczny takie dokumenty jak np. „xml” czy „txt”, które zawierają treści tj. obrazy, teksty, tabele itp. Autor projektu użył tej biblioteki, aby po zakończeniu rejestracji czasów pomiędzy wydaniem strzału a trafianiem pocisku w tarczę wszystkie wyniki były umieszczane w pliku PDF (maksymalna wartość czasowa strzału, minimalna wartość czasowa strzału, refleks strzelca rozumiany jako średnia wartość czasów wszystkich strzałów itd.). Cały kod odpowiedzialny za tworzenie pliku, za pomocą tej biblioteki, jest umieszczony w funkcji programu „create\_pdf()” w formularzu aplikacji „menu\_refleks”. Poniżej w formie graficznej przedstawiony został fragment kodu tworzącego taki dokument. [4]



Rys. 3.2. Przykład użycia klasy odpowiadającej za tworzenie dokumentu napisana w języku C#.

### 3.1.2. Media – Biblioteka C#

Biblioteka „Media” odpowiada za odtwarzanie plików dźwiękowych w formacie „wav”. W tym projekcie inżynierskim została ona wykorzystana do odtworzenia dźwięku sygnalizującego osobie wykonującej strzał moment rozpoczęcia oraz zakończenia pomiaru. Dźwięk jest odtwarzany 5 sekund przed zapaleniem się diody sygnalizującej

rozpoczęcie oraz po 3 minutach sygnalizując zakończenie pomiaru refleksu strzelca. Kod programu, którego używa ta biblioteka znajduje się w funkcji „voice\_sound” w głównym formularzu aplikacji „menu\_refleks. [5]

### **3.1.3. Data.SqlClient – Biblioteka C#**

„Data.SqlClient” to biblioteka platformy programistycznej .NET odpowiadającej za obsługę „Microsoft SQL Server”, inaczej system zarządzający danymi. Platforma ta działa na zasadzie klient-serwer używając tzw. zapytań, stanowiących główny element języka DML (Data Manipulation Language) wchodzącego w język SQL i manipulującego danymi. Wysyłając zapytanie z aplikacji formularza systemu Windows należy jednak pamiętać, że w przypadku błędnego zapytania nie otrzymamy odpowiedzi zwrotnej o błędzie. Zapytania używane w projekcie umieszczone są w wielu fragmentach kodu umożliwiając przez to np. usuwanie danych z bazy danych lub modyfikację zawartych w niej informacji. Dzięki klasie „SqlConnection” możliwe jest połączenie serwera bazy danych z główną aplikacją projektu. [6]

### **3.1.4. IO – Biblioteka C#**

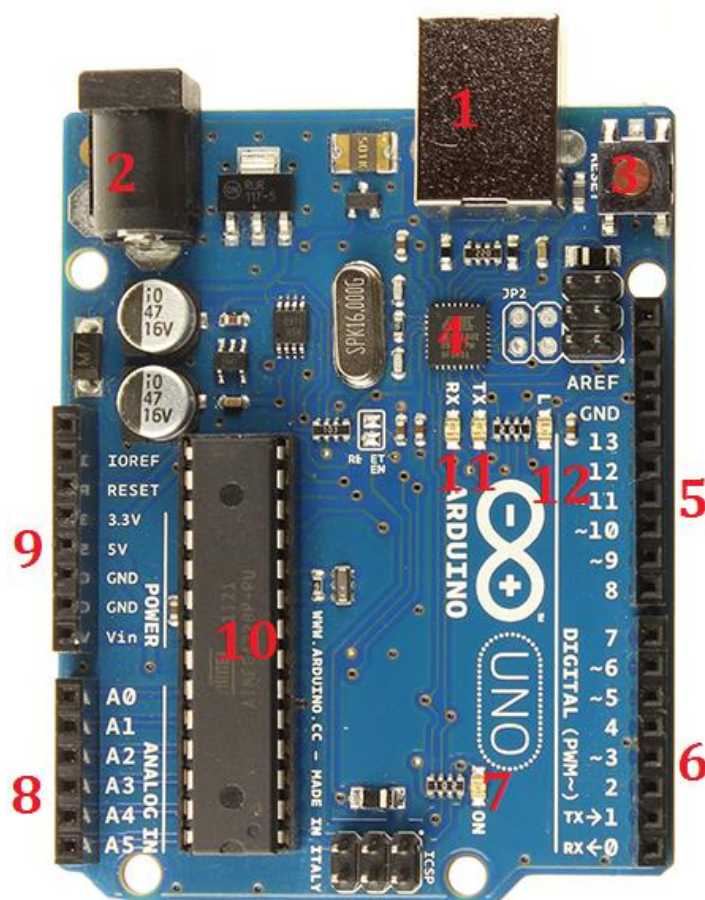
Ta biblioteka odpowiada za posiadanie typów zmiennych, pomagających w zapisie i w odczycie do pliku jak i strumienia danych. Biblioteki „IO” użyto do usuwania lokalnych plików PDF, co zapobiega zbędnemu przeciążaniu pamięci komputera. [7]

## **3.2. MySQL**

Jest to system zarządzający bazami danych. W projekcie użyto tego systemu zarówno do wydobycia z bazy danych MySQL informacji, jak i magazynowania nowych danych, dotyczących broni oraz wyników pomiaru refleksu strzelca. System po dziś dzień jest rozwijany dzięki firmie Oracle. W projekcie inżynierskim użyto tej bazy danych ze względu na jej niezawodność, wydajność oraz łatwość obsługi przez użytkownika. Ponadto posiada ona tytuł najbardziej znanego i używanego środowiska baz danych dla małych projektów, jakim jest również projekt układu do badania refleksu snajpera strzelającego z broni typu ASG. MySQL jest rodzajem systemu „otwartego oprogramowania” tylko i wyłącznie do użytku własnego (nie komercyjnego) i opiera się na licencji GPL-GNU (General Public Licence). [8]

### 3.3. Arduino

Arduino to elektroniczna platforma typu „open source” używająca prostego w obsłudze sprzętu i oprogramowania. Platforma ta daje możliwość odczytu wartości wejścia z czujników. W przypadku tego projektu konieczne jest odczytanie sygnałów pomiarowych z akcelerometru, który używając komunikacji UART, dzięki przewodowi USB nawiązuje połączenie z PC i wysyła gotowe dane do odbioru na komputerze. Ten, w zależności od odebranej wartości z czujnika akcelerometru, powoduje włączenie się diody lub jej wyłączenie, wysyłając jednocześnie informację zwrotną o stanie diody do środowiska programistycznego Arduino (IDE). To oprogramowanie posiada wiele bibliotek i usprawnień sprzętowych, dlatego nie wymaga dużej znajomości elektroniki, co ułatwia programowanie na komputerze osobistym w systemie Windows. W swoim projekcie autor wykorzystał to oprogramowanie do obsługi czujnika przyspieszenia i diody. Poniżej przedstawiono płytke Arduino Uno oraz opisano najważniejsze zawarte w niej elementy: [9]



Rys. 3.3. Widok płytki Arduino Uno. [10]

Opis poszczególnych elementów płytki Arduino Uno:

**Nr 1.** Złącze USB odpowiadające za zasilanie oraz za komunikację z komputerem. Obie funkcje zostały wykorzystane w projekcie.

**Nr 2.** Podzespół również odpowiadający za zasilanie układu napięciem o wartości od 7V do 12V. Jest on wykorzystywany w przypadku niekorzystania ze złącza USB.

**Nr 3.** Przycisk odpowiadający za reset płytki Arduino Uno.

**Nr 4.** Mikrokontroler komunikujący się z komputerem poprzez port USB. Mikrokontroler został wykorzystany w programie do wysyłania danych z akcelerometru oraz do odbierania wiadomości z komputera informujących o włączeniu lub wyłączeniu się diody.

**Nr 5 i 6.** Piny (złącza), które mogą służyć jako cyfrowe wejścia lub wyjścia z możliwością zaprogramowania. W projekcie inżynierskim do pinu nr „8” została podłączona dioda sygnalizująca możliwość oddania strzału.

**Nr 7.** Dioda sygnalizująca pracę układu.

**Nr 8.** Piny – wejścia analogowe. W projekcie inżynierskim skorzystano z następujących pinów oznaczonych nr „8”.: pin nr „A4”, pin nr „A5”. Pin „A5” podłączony został do pinu „SCL” (Serial Clock) akcelerometru. Pinu „A4” użyto do odczytania linii danych SDA (Serial Data) czujnika przyspieszenia. Tylko te dwa piny obsługują magistralę I<sup>2</sup>C (dwukierunkowa magistrala służąca do przesyłania danych w urządzeniach elektronicznych). Odpowiada ona za nawiązanie komunikacji pomiędzy urządzeniami podrzędnymi zwanymi „slave” a urządzeniami głównymi zwanymi „master” oraz za sterowanie tymi układami.

**Nr 9.** Piny zasilania, do których został podłączony układ sprawdzający wartość przyspieszenia czujnika akcelerometrycznego. Użyto do tego pinu „GND” (uziemia dla diody i akcelerometru) oraz pinu „5V” (zasilanie akcelerometru).

**Nr 10.** Podzespół o nazwie AVR (Advanced Virtual Risc) pełniący funkcję mikrokontrolera typu ATmega328. Jest najważniejszym elementem w układzie Arduino Uno, wykonującym 16 MLN operacji na sekundę.

**Nr 11.** Diody LED informujące o dwukierunkowej transmisji danych systemu. Rola diod polega na sygnalizowaniu częstotliwości z jaką przesyłana jest transmisja.

**Nr 12.** Dioda sygnalizująca informująca użytkownika o gotowości układu do użycia.

### **3.3.1. Układ MPU6050 – Biblioteka Arduino**

Układ elektroniczny MPU6050 składa się z: trójosiowego akcelerometru, termometru oraz trójosiowego żyroskopu. Układ umożliwia stabilny pomiar danych dotyczących położenia czujników względem ziemi oraz prędkość wyjściową o częstotliwości nie większej niż 1000Hz. MPU6050 używa I<sup>2</sup>C do komunikowania się z mikrokontrolerem. Zaletą tego układu jest przeliczanie gotowych danych, z trzech czujników jednocześnie, na konkretne położenie. Proces ten jest możliwy dzięki użyciu jednostki DMP (Digital Motion Procesor). Układ MPU6050 posiada również 16 bitowy przetwornik, co daje mu przewagę nad 13 bitowym przetwornikiem modelu ADXL345, który również był brany pod uwagę w trakcie doboru układu mierzącego położenie czujników. Dostęp do zasobów darmowej biblioteki, udostępnionej przez Korneliusza Jarzębskiego, pozwala na odczytywanie wartości przyspieszenia akcelerometru. Jest to możliwe dzięki funkcji „readRawAccel” odczytującej surowe dane z czujników osi x, y i z akcelerometru, oraz funkcji „readNormalizeAccel” odpowiedzialnej za ukazywanie znormalizowanych wartości gotowych do przesłania przez port USB do aplikacji znajdującej się na komputerze. [11]

### **3.4. Środowisko programistyczne**

Zintegrowane programistyczne środowisko (IDE) to pojedynczy program lub zespół programów wspólnie tworzących tzw. środowisko programistyczne umożliwiające tworzenie, modyfikowanie, konserwowanie oraz debuggowanie oprogramowania. Poprzez określenie „zintegrowane środowisko programistyczne” mamy na myśli, że pojedyncza aplikacja może udostępnić wielu funkcji występujących w innym oprogramowaniu. Zdecydowana większość przypadków dotyczy funkcji zajmujących się edytowaniem kodu źródłowego, budowaniem referencji lub zasobów, a także kompilacją kodu. Praktyczne zastosowanie IDE obejmuje również korzystanie z przygotowanych bibliotek oraz odpowiada za budowanie różnych baz i komponentów. [12]

#### **3.4.1. Microsoft Visual Studio**

Jest to aplikacja wyższego poziomu stworzona przez firmę Microsoft. Używana jest do tworzenia oprogramowania przez wbudowany interfejs graficzny, przeznaczony dla użytkownika pozwalający na użycie narzędzi graficznych systemu Microsoft Windows. Program funkcjonuje dzięki zasadzie „zdarzeń wspomaganych” za pomocą .Net Framework. Aplikacja zaczyna działać bezpośrednio po wykryciu ingerencji użytkow-

nika, a następnie realizuje kod znajdujący się pod ukrytymi polami tekstu, czasomierzami itd. Metoda ta dobrze sprawdza się we współpracy z oprogramowaniem komunikującym się z UARTE – em oraz korzystającym z bazy danych MySQL. Wykorzystując eksploratora systemu Windows w prosty sposób możemy odczytać wiadomość przychodzącą z portu USB lub znaleźć interesującą użytkownika bazę danych. Z poziomu aplikacji Microsoft Visual Studio można obejrzeć czy wyselekcjonowany plik jest prawidłowy, a wykorzystując przyciski graficzne wgrać poprawki na programowy magazyn danych. W wyniku wyżej wymienionych korzyści w tym projekcie inżynierskim wykorzystano do opracowania oprogramowania aplikację Microsoft Visual Studio, używając Formularza Windows. [13]

### **3.5. Środowisko CAD**

Programowanie w środowisku CAD (Computer Aided Design) to inaczej programowanie przy wykorzystaniu projektowania wspomagane komputerowo. Technologia ta jest wykorzystywana, między innymi przez inżynierów, do tworzenia modeli cyfrowych konstrukcji dwuwymiarowych 2D i trójwymiarowych 3D oraz dokumentacji konstrukcyjnej na podstawie modeli 2D i 3D. Wykorzystywana jest również do drukowania modeli przy użyciu drukarki 3D. Środowisko CAD usprawnia proces generowania modelu konstrukcyjnego poprzez przysyłanie szczegółowych danych dotyczących materiału, tolerancji czy wymiarów modelu. Zaletami tego programu jest przede wszystkim możliwość korzystania z dużej ilości narzędzi wykorzystywanych w procesie projektowania, duży zasób bibliotek materiałowych (umożliwiających użytkownikowi nadanie znaczenia materiałowego poszczególnym elementom projektu) oraz możliwość modyfikacji i ulepszania powstałych już elementów w przypadku potrzeby zastosowania późniejszych zmian. Wszystkie wyżej wymienione zalety środowiska CAD spełniały podstawowe oczekiwania autora projektu dotyczące opracowania projektu 3D układu mechanicznego do badania refleksu snajpera strzelającego z broni typu ASG oraz wykonania na jego podstawie rysunku technicznego. [14]

#### **3.5.1. Autodesk Inventor Professional**

Autodesk Inventor jest programem typu CAD. Jest aplikacją wspomagającą studentów poprzez darmowy dostęp do oprogramowania. Program Autodesk Inventor ma zastosowanie w elektryce, mechanice oraz budownictwie. W ramach projektu inżynierskiego został użyty do odwzorowania w trójwymiarowej konstrukcji mechanicznej

oraz sprawdzenia wymiarów układu i sposobu zamontowania profilu o przekroju kwadratowym z płytą wiórową. Następnie dzięki możliwości przekształcenia gotowego modelu z formatu „.ipt” do formatu „.idw” , został sporządzony rysunek techniczny na bazie modelu 3D. Na podstawie ww. rysunku technicznego w zakładzie rzemieślniczym Trepko Aseptic zostały wykonane elementy układu mechanicznego tj. : profil o przekroju kwadratowym, płaskownik oraz tarcza. Autor projektu zdecydował się na użycie programu Autodesk Inventor Professional, ponieważ przy tworzeniu prostych modeli nie obciąża w znacznym stopniu sprzętu oraz jest intuicyjny w zakresie korzystania z interfejsu. [15]

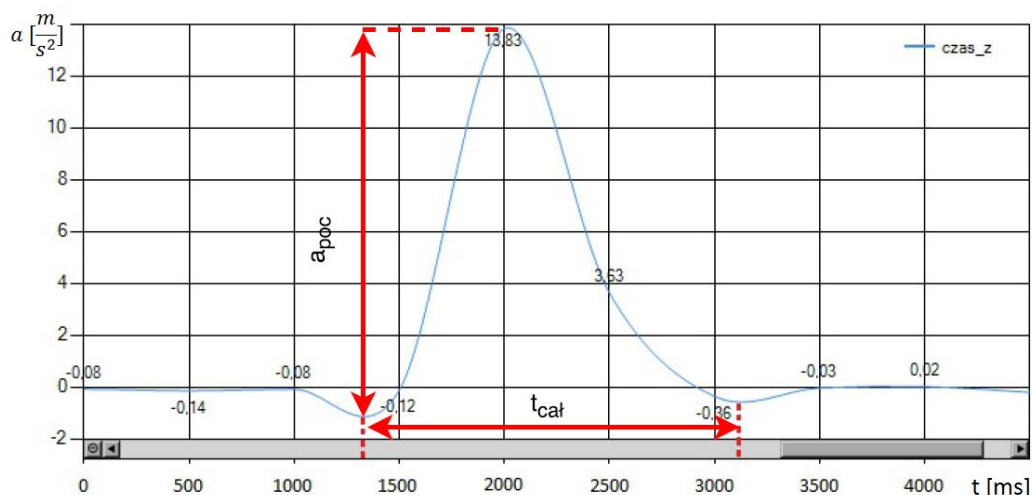
### **3.5.2. FEM – Autodesk Inventor**

FEM (Finite Element Method) czyli metoda elementów skończonych, to numeryczna metoda rozwiązywania równań różniczkowych. Polega ona na określeniu warunków brzegowych, a następnie na podzieleniu całego elementu na mniejsze i prostsze części. Następnie na mniejszych elementach wykonywane są obliczenia, które opracowując system równań analizują zaistniały problem.

Autodesk Inventor Professional zawiera środowisko programistyczne do analizy naprężeń, co też wykorzystano przy wyborze materiału na tarczę strzelniczą poprzez sprawdzenie poziomu odkształcenia oraz współczynnika bezpieczeństwa tarczy po uderzeniu w nią pocisku. Wartość siły z jaką pocisk uderza w tarczę jest obliczana na podstawie wzoru przedstawionego pod tekstem. Na poniższych rysunkach przedstawiono i opisano przykładowe testy badające wytrzymałość tarczy oraz określające wychylenie tarczy z położenia początkowego (przed uderzeniem pocisku).

Na początku wyznaczono wartość maksymalnego przyspieszenia tarczy, o masie 191 g, w momencie uderzenia pocisku w tarczę. W tym celu oddano kilka strzałów poprzez to sprawdzając, w jakim czasie nastąpi przekazanie całej energii tarczy przy maksymalnej prędkości wylotowej 152 m/s oraz masy kulki 0,25 g. Następnie na podstawie odczytanej ww. wartości maksymalnego przyspieszenia dokonano obliczeń siły jaka działa na tarczę. Wyniki obliczeń wykorzystano podczas symulacji.





Rys. 3.4. Wykres zależności przyspieszenia „a” w czasie „t” dla masy pocisku 0,25 g oraz prędkości pocisku 152 m/s podczas rejestracji pocisku.

Na Rys. 3.4 widoczne jest znaczne zwiększenie przyspieszenia o wartości  $14 \text{ m/s}^2$  (oznaczone „ $a_{poc}$ ”), które następuje w momencie trafienia pocisku w tarczę. „ $t_{cał}$ ” natomiast jest oznaczeniem czasu pomiędzy trafieniem, a wytłumieniem ruchu tarczy. Po kilkukrotnie oddanych strzałach rejestrowane przyspieszenie wynosiło ok.  $14 \text{ m/s}^2$ . Taką wartość przyjęto podczas obliczeń

Następnie zgodnie z poniższym wzorem określono siłę pocisku:

$$F = a_{poc} * m_{tar} [N]$$

Gdzie:

$m_{tar}$  - masa tarczy

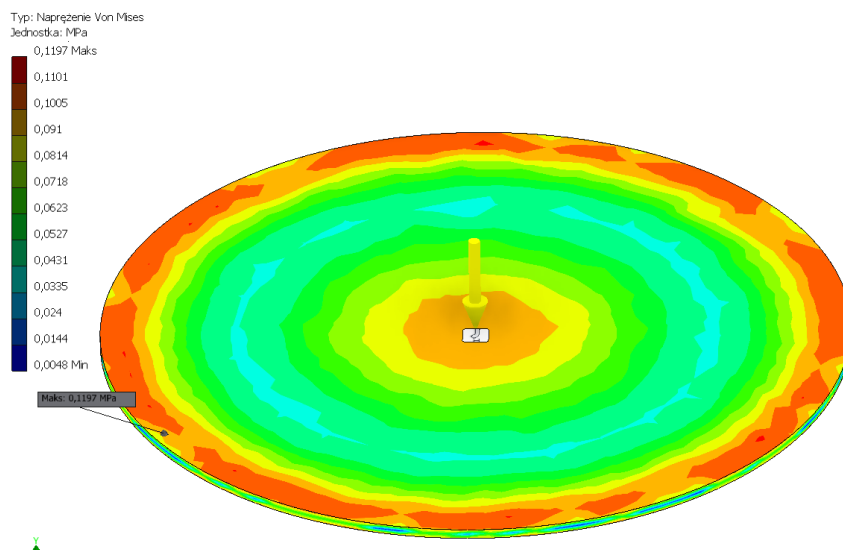
$a_{poc}$  - przyspieszenie tarczy

Po podstawieniu wartości wzór wygląda następująco:

$$F = 14 \frac{m}{s^2} * 1,91 kg * 10^{-1}$$

$$F \approx 2,67 N$$

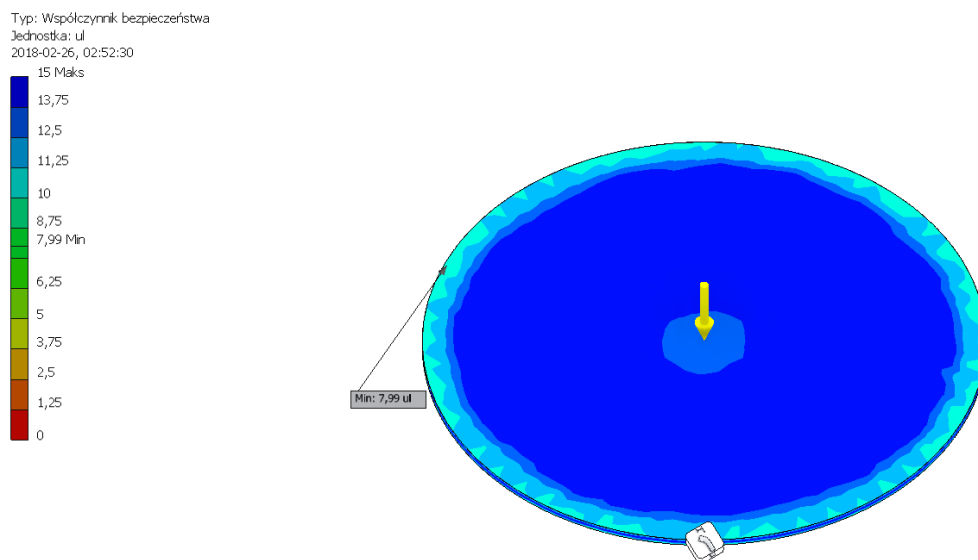
Podczas symulacji naprężeń zadawano siłę nie mniejszą niż  $2,7 \text{ N}$ . Wiązania nieruchome stanowią ściany boczne tarczy, co zostało przedstawione na Rys. 3.5:



**Rys. 3.5.** Wyniki symulacji naprężeń wytworzonych w tarczy (wykonanej ze stali nierdzewnej o grubości 2 mm i średnicy 150 mm) przez działającą na nią siłę o wartości 2,7 N.

Wynikiem przeprowadzonej symulacji jest maksymalna wartość naprężeń tarczy wynosząca 0,1197 MPa. Naprężenia te nie powodują większych uszkodzeń.

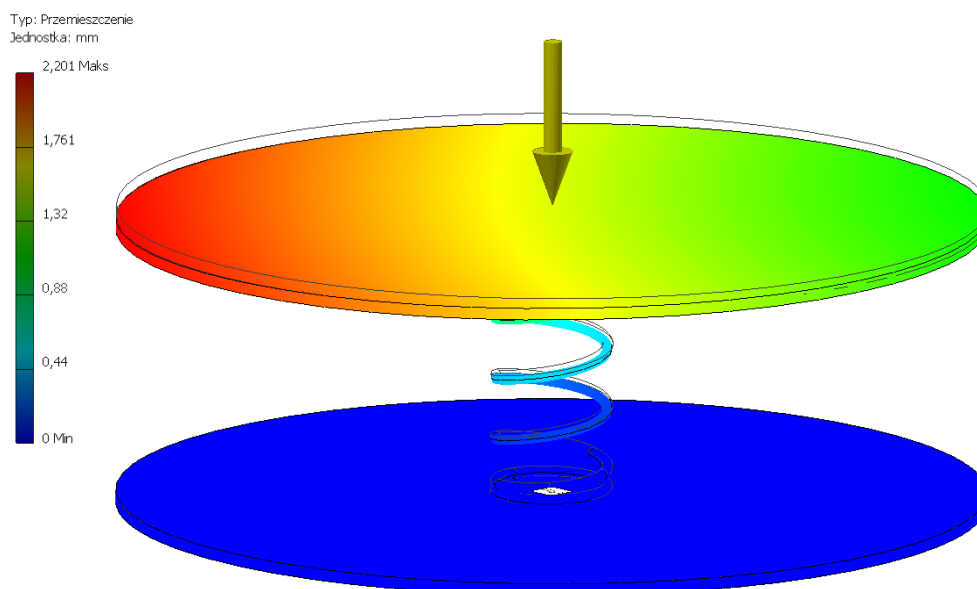
Następna symulacja dotyczy obliczenia współczynnika bezpieczeństwa tarczy:



**Rys. 3.6.** Wyniki symulacji przemieszczeń tarczy (wykonanej ze stali grubości 2 mm i średnicy 150 mm) powstałych przez oddziaływanie na nią siły o wartości 500 N (Newton), wyrażone w  $\sigma_n / \sigma_{dop}$  (Naprężenie niebezpieczne/Naprężenie dopuszczalne).

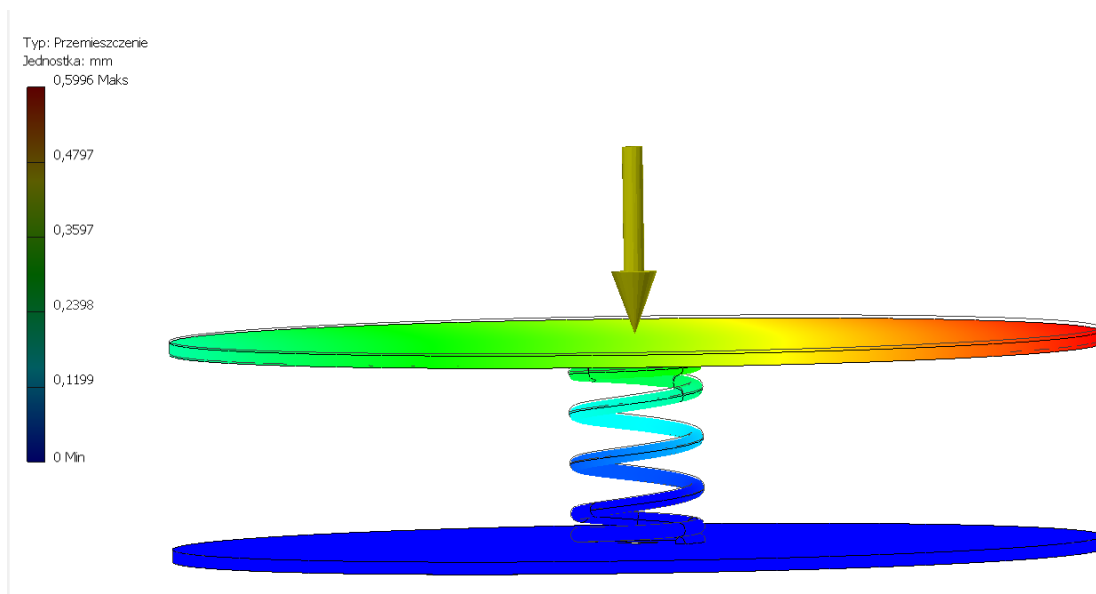
Przeprowadzając przykładowe obliczenia współczynnika bezpieczeństwa, zmiany tej wartości można zauważyć dopiero po zastosowaniu siły 500 N. Współczynnik ma wartość  $7,89 \sigma_n / \sigma_{dop}$ . Oznacza to, że naprężenie dopuszczalne nie przekracza naprężenia niebezpiecznego tym samym nie powodując jego zniszczenia. Jeśli współczynnik naprężeń znajdowałby się blisko wartości  $1 \sigma_n / \sigma_{dop}$  to konstrukcja uległaby uszkodzeniu.

Rys. 3.7 i 3.8 przedstawiają przemieszczenie się tarczy, przymocowanej do sprężyny (wykonanej z narzędzia „sprężyna naciskowa” w zakładce projekt, który tworzy heliakalną sprężynę śrubową, na podstawie określonych wymiarów), w momencie uderzenia w nią pocisku. Na jednej z tarcz (dolnej) użyto wiązania nieruchomego podczas tworzenia symulacji.



**Rys. 3.7. Wyniki symulacji przemieszczenia tarczy( wykonanej z aluminium o grubości 2 mm i średnicy 15 mm), przeprowadzonej w układzie ze sprężyną (o współczynniku sztywności 1,99 N/mm) , poprzez oddziaływanie na nią siły o wartości 2,6 N.**

Wynik przemieszczenia wynosi (w przybliżeniu) ok. 2,2 mm, co jest wartością wystarczającą dla akcelometru znajdującego się pod tarczą do rejestracji chwilowego przyspieszenia.



**Rys. 3.8.** Wyniki symulacji przemieszczenia tarczy( wykonanej z aluminium o grubości 2 mm i średnicy 15 mm), przeprowadzonej na układzie ze sprężyną (o współczynniku sztywności 8,33 N/mm) , poprzez oddziaływanie na nią siły o wartości 2,6 N.

Podczas tej symulacji przemieszczenie wyniosło ok. 0,60 mm. Jest to wartość, która może być zbyt mała, aby akcelerometr mógł zarejestrować jako trafienie.

**Kreator sprężyn naciskowych**

**Projekt** **f<sub>s</sub>** **Obliczenia**

Obliczenie wytrzymałościowe sprężyny  
Kontrola obliczenia sprężyny

Opcje obliczeń  
Typ projektu  
F<sub>r</sub> Wymiary montażowe --> d, L<sub>0</sub>, n, D

Metoda korekty krzywizny naprężenia  
Brak korekcy

Projekt wymiarów montażowych  
Projekt wszystkich wymiarów montażowych L<sub>1</sub>, L<sub>0</sub>, H

**Obciążenie**  
Minimalne obciążenie F<sub>1</sub> 2,000 N  
Maksymalne obciążenie F<sub>8</sub> 6,000 N  
Obciążenie robocze F 3,000 N

**Wymiary**  
Średnica drutu d 1,7 mm  
Średnica zewnętrzna D<sub>1</sub> 21 mm  
Długość swobodna sprężyny L<sub>0</sub> 49 mm

**Zwoje sprężyny**  
Zaokrąglenie liczby zwojów 1  
Zwoje aktywne n 5 ul

**Materiał sprężyny**  
Materiał użytkownika  
Granica napr. rozciągającego σ<sub>ult</sub> 1860,000 MPa  
Dopuszczalne graniczne naprężenie skręcające τ<sub>A</sub> 930,000 MPa  
Moduł elastyczności w ścisku G 68500,000 MPa  
Gęstość ρ 7850 kg/m<sup>3</sup>  
Współczynnik wykorzystania materiału us 0,900 ul

**Kontrola wyboczenia**  
Typ sprężyny  
Prowadzone osadzanie - równobieżnie obrabione płaszczyzny przylegające

**Obciążenie zmęczeniowe**  
Sprężyny niekulowane  
Trwałość sprężyny w tysiącach ugięć N > 10000  
Współczynnik bezpieczeństwa k<sub>f</sub> 1,200 ul

**Wymiary montażowe sprężyny**  
H, L<sub>1</sub> --> L<sub>0</sub>  
Długość przy minimalnym obciążeniu L<sub>1</sub> 47,995 mm  
Długość przy maksymalnym obciążeniu L<sub>8</sub> 45,984 mm  
Skok roboczy H 2,011 mm  
Długość przy obciążeniu roboczym L<sub>w</sub> 47,492 mm

**Wyniki**  
a 7,335 mm  
t 9,035 mm  
K<sub>w</sub> 1,000 ul  
K 1,990 N/mm  
s<sub>1</sub> 1,005 mm  
s<sub>8</sub> 3,016 mm  
s<sub>9</sub> 36,675 mm  
L<sub>minf</sub> 14,680 mm  
L<sub>9</sub> 12,325 mm  
F<sub>9</sub> 72,967 N  
τ<sub>1</sub> 20,007 MPa  
τ<sub>8</sub> 60,021 MPa  
τ<sub>9</sub> 729,920 MPa  
v 20,427 mps  
f 303,445 Hz  
W<sub>B</sub> 0,009 J  
l 463,200 mm  
m 0,008 kg

Oblicz OK Anuluj >>

**Rys. 3.9.** Kreator sprężyny naciskowej z określonymi wymiarami.

Na Rys. 3.9 przedstawiono parametry dla sprężyny wykorzystanej do symulacji uderzenia pocisku (Rys. 3.7) oraz użytej w pracy inżynierskiej. Kreator pozwala

na dopasowanie sprężyny do obciążeń roboczych znajdujących się w układzie. Gdyby wybrana sprężyna nie spełniała tych warunków, to pojawia się komunikat o błędzie projektu sprężyny np. "długość przy maksymalnym obciążeniu jest mniejsza niż 0".

### 3.6. Obliczanie sprężyny śrubowej naciskowej

Sprężyna naciskowa wykonana jest ze sprężystej stali i budową przypomina cewkę. Pochłania ona siłę lub zapewnia opór podczas działania na nią siły i może być używana między innymi jako amortyzator lub generator siły. Autor projektu wykorzystał zdolność pochłaniania siły przez sprężynę jako amortyzatora. Jest to niezbędne do rejestrowania wartości przyspieszenia przez akcelerometr. W celu obliczenia współczynnika sprężystości sprężyny naciskowej zastosowano wzory znajdujące się poniżej. Na początku należało określić ciężar ciała za pomocą następującego wzoru:

$$Q = m * g [N]$$

gdzie:

**m**- masa ciała zawieszonego na sprężynie

**g** – przyspieszenie ziemskie

$$Q = 0,500 \text{ kg} * 9,81 \frac{N}{kg}$$

$$Q = 4,905 \text{ N}$$

W dalszych obliczeniach zastosowano wzór obliczania współczynnika sprężystości przedstawiony poniżej:

$$k = \frac{m * g}{x_0} \left[ \frac{N}{mm} \right]$$

gdzie:

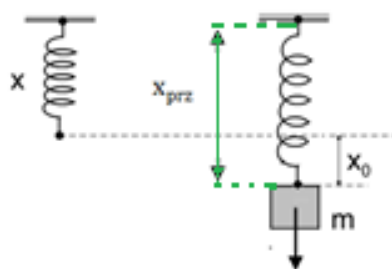
$x_0$  – długość rozciągnięcia pod wpływem ciężaru

Jako, że wartość długości  $x_0$  nie jest znana należy obliczyć ją zgodnie z poniższym wzorem:

$$x_0 = x - x_{prz} [mm]$$

$x$ - długość sprężyny bez obciążenia

$x_{prz}$  – długość sprężyny po obciążeniu masą 0,5 kg



**Rys. 3.10.** Przedstawienie graficzne sprężyny nieobciążonej(po lewej stronie) oraz sprężyny obciążonej, wychylonej z położenia równowagi (po prawej stronie).

Po obciążeniu sprężyny i zmierzeniu jej odkształcenia wartość długości  $x_0$  określa się następująco:

$$x_0 = 51,6 \text{ mm} - 49 \text{ mm}$$

$$x_0 = 2,6 \text{ mm}$$

Posiadając wszystkie niezbędne wartości można obliczyć współczynnik sprężystości dla sprężyny śrubowej.

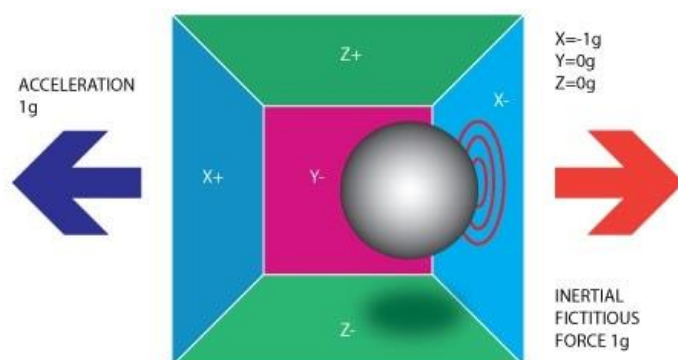
$$k = \frac{4,905 \text{ N}}{2,6 \text{ mm}}$$

$$k = 1,86 \frac{\text{N}}{\text{mm}}$$

Analogicznie obliczono współczynniki sprężystości dla pozostałych dostępnych sprężyn. Wyniki obliczeń umieszczono w Tab. 4.1.

### **3.7. Akcelerometr**

Akcelerometr, zwany również przyśpieszeniomierzem, jest czujnikiem głównie mierzącym przyśpieszenie (może również badać położenie oraz prędkość). Do wykonania obliczeń akcelerometr używa wektora przyśpieszenia. Jego zmiana następuje poprzez efekt piezoelektryczny.

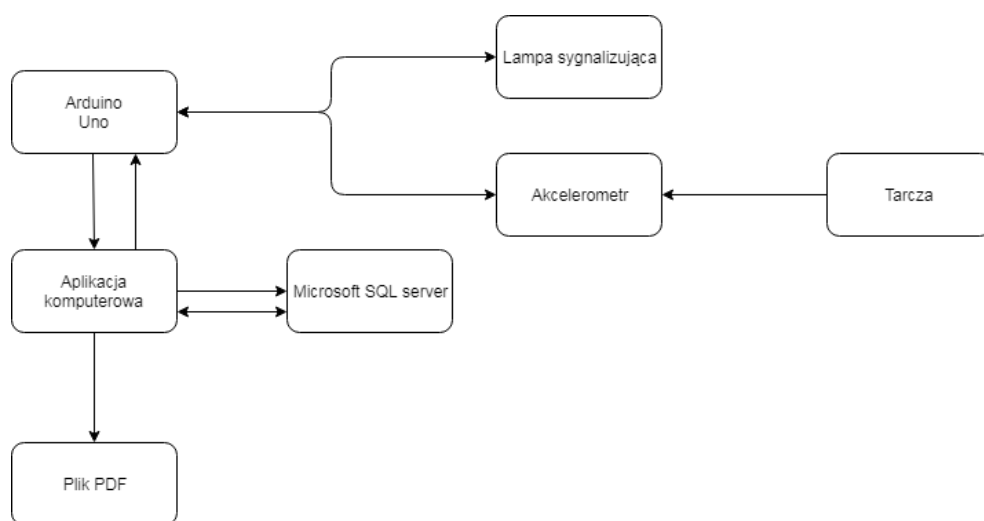


**Rys. 3.11. Graficzne przedstawienie zasady działania piezoelektrycznego akcelerometru. [16]**

Na powyższym rysunku widoczna jest kula wewnątrz prostopadłościanu. Jego ściany są odwzorowaniem osi z, y i x (w formacie 3D) i wykonane są z kryształów piezoelektrycznych. W momencie nadania prędkości kuli porusza się ona w kierunku przeciwnym przez działającą na nią siłę bezwładności. W skutek tego uderza z określoną siłą w ścianę prostopadłościanu, a moment zderzenia kuli ze ścianą prostopadłościanu jest momentem wytworzenia się prądu elektrycznego. Na podstawie wartości wytworzonego prądu elektrycznego można określić kierunek nachylenia i wychYLENIA czujnika. W projekcie inżynierskim wyżej opisany efekt został wykorzystany do rejestracji wartości przyspieszenia przetwornika przyspieszenia umieszczonego pod tarczą. [16]

## 4. Wdrożenie

Ten rozdział poświęcony jest przedstawieniu układu mechanicznego i elektronicznego oraz omówieniu najważniejszych elementów układu wraz ze sposobem ich działania. Opracowano tu również pokrótce przedstawiony wygląd interfejsu, sposób implementacji opracowanego systemu oraz sposób działania najważniejszych funkcji użytych w programie. Na rysunku zamieszczonym poniżej ukazano w sposób graficzny zależności występujące pomiędzy poszczególnymi elementami układu opracowanego w projekcie:



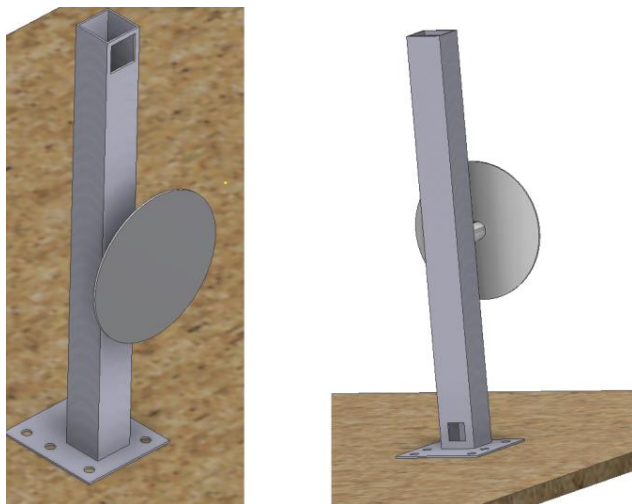
**Rys. 4.1. Graficzne przedstawienie zależności pomiędzy układem elektronicznym, mechanicznym i programem znajdującym się na komputerze stacjonarnym.**

Tarcza znajdująca się na powyższym schemacie, po nadaniu jej przyspieszenia, przekazuje przez zmianę położenia sprężyny informację do akcelerometru. Zarówno nadanie przyspieszenia tarczy, jak i zmiana położenia sprężyny jest następstwem uderzenia w tarczę pocisku wystrzelonego z broni ASG. Następnie do płytki Arduino Uno wysyła odczyt wartości z akcelerometru, które przetwarza opracowana aplikacja. Tam dokonuje się analiza ww. wartości w kodzie aplikacji badającej refleks strzelca. Jeśli wartość przyspieszenia ulegnie zwiększeniu, to aplikacja komputerowa wysyła polecenie do płytki Arduino Uno o wyłączenia diody. Po zakończeniu rejestracji pomiarów niezbędnych do obliczenia refleksu strzelca poprzez połączenie aplikacji komputerowej z bazą danych wysyłane są do niej wyniki pomiaru. Dzieje się tak po oddaniu minimum 10 strzałów. Na koniec tego procesu udostępniany jest plik zawierający wyniki ww. pomiaru.



#### **4.1. Układ mechaniczny**

W podpunkcie 4.1 autor projektu krótko opisał, opracowany w programie Autodesk Inventor Professional, model układu mechanicznego oraz rysunek techniczny na podstawie którego wykonano układ rzeczywisty. W kolejnych podpunktach został omówiony dobór sprężyny oraz pozostałych materiałów wykorzystanych do wykonania ww. układu.



**Rys. 4.2.** Graficzne przedstawienie modelu opracowanego układu mechanicznego [widok z przodu (po lewej stronie) i z tyłu (po prawej stronie)]. [17]

Na podstawie powyższego modelu opracowanego w Autodesk Inventor Professional sporządzono rysunek techniczny tarczy oraz profilu o przekroju kwadratowym z płaskownikiem. Następnie część mechaniczną wykonano w zakładzie „Trepko Aseptic” w sposób widoczny na Rys. 4.2. Kolejnym krokiem jest złączenie sprężyny ze stalowym profilem oraz z aluminiową tarczą. Proces ten stał się możliwy przez zastosowane spoiwo metaliczne „Poxipol”. Ostatnim etapem jest umieszczenie i przymocowanie konstrukcji do płyty wiórowej za pomocą śruby z sześciokątnym łbem o standardzie M8 x 30 oraz połączenie z drugiej strony podkładką płaską. Na nią nałożono podkładkę kontruującą w celu zmniejszenia prawdopodobieństwa poluzowania się mocowania oraz nakrętkę zabezpieczającą przed rozkręcaniem się połączenia śrubowego.

#### 4.1.1. Dobór sprężyny i materiału

Podpunkt ten zawiera tabelę ze spisem materiałów, na których przeprowadzano symulację wytrzymałości. Proces ten był niezbędny przy doborze odpowiedniego materiału, z którego wykonane były poszczególne elementy wchodzące w skład układu mechanicznego. Wyniki umieszczone w tabeli zostały obliczone na podstawie symulacji przedstawionych na Rys. 3.5 i 3.6. W drugiej tabeli zostały umieszczone wartości wymiarów oraz podstawowych parametrów sprężyn na podstawie punktu 3.7. Poniżej zamieszczono tabele wraz z objaśnieniem kryteriów doboru danego materiału.

**Tab. 4.1. Wytrzymałość materiałów poddanych symulacji w programie Autodesk Inventor Professional.**

L.P	Materiał	$\sigma_{\max}$ , MPa	$k, \sigma_n / \sigma_{\text{dop}}$	$m$ , g	$\rho$ , g/cm <sup>3</sup>	cena, zł/kg
1	Stal nierdzewna	0,1381	15	282,60	8,00	0,6
2	Aluminium	0,1355	15	95,38	2,70	3,7
3	Mosiądz	0,1354	15	299,20	8,47	13,5
4	Żeliwo	0,1389	15	252,57	7,15	0,75
5	Miedź	0,1341	15	299,91	8,49	18

Gdzie:

$\sigma_{\max}$  – naprężenia maksymalne w megapaskalach

$k$  - współczynnik bezpieczeństwa

$m$  - masa wyrażona w gramach

$\rho$  – gęstość materiału wyrażona w gramach na centymetr sześcienny

**cena** - wyrażona złotych za kilogram

Z powyższej tabeli można odczytać, że wszystkie materiały spełniały warunki wytrzymałościowe (dotyczące) doboru tarczy dla siły o wartości 3 N. Materiał wybrany do wykonania tarczy to aluminium. Autor projektu wybrał właśnie ten materiał, ponieważ jako jedyny spełniał on założenie, mówiące o zależności pomiędzy ciężarem materiału, z jakiego wykonana jest tarcza, a siłą potrzebną do wprowadzenia w ruch tarczy połączonej ze sprężyną.

**Tab. 4.2. Wymiary sprężyn oraz określone parametry na podstawie punktu 3.7.**

L.P	$L_0$ , mm	$D_i$ , mm	$d$ , mm	$D_e$ , mm	$n$ , ilość	$k$ , N/mm
1	65,7	12,50	1,5	15,50	15	1,225
2	49,0	17,60	1,7	21,00	5	1,990
3	38,1	6,15	1,5	9,10	12	8,069
4	15,7	11,85	1,7	15,20	3	10,365
5	31,8	16,90	2,3	21,60	4	8,333

Gdzie:

$L_0$  - długość sprężyny nieobciążonej wyrażona w milimetrach

$D_i$  - średnica wewnętrzna sprężyny wyrażona w milimetrach

$d$  - średnica drutu, z którego wykonano sprężynę, wyrażona w milimetrach

$D_e$  - średnica zewnętrzna sprężyny wyrażona w milimetrach

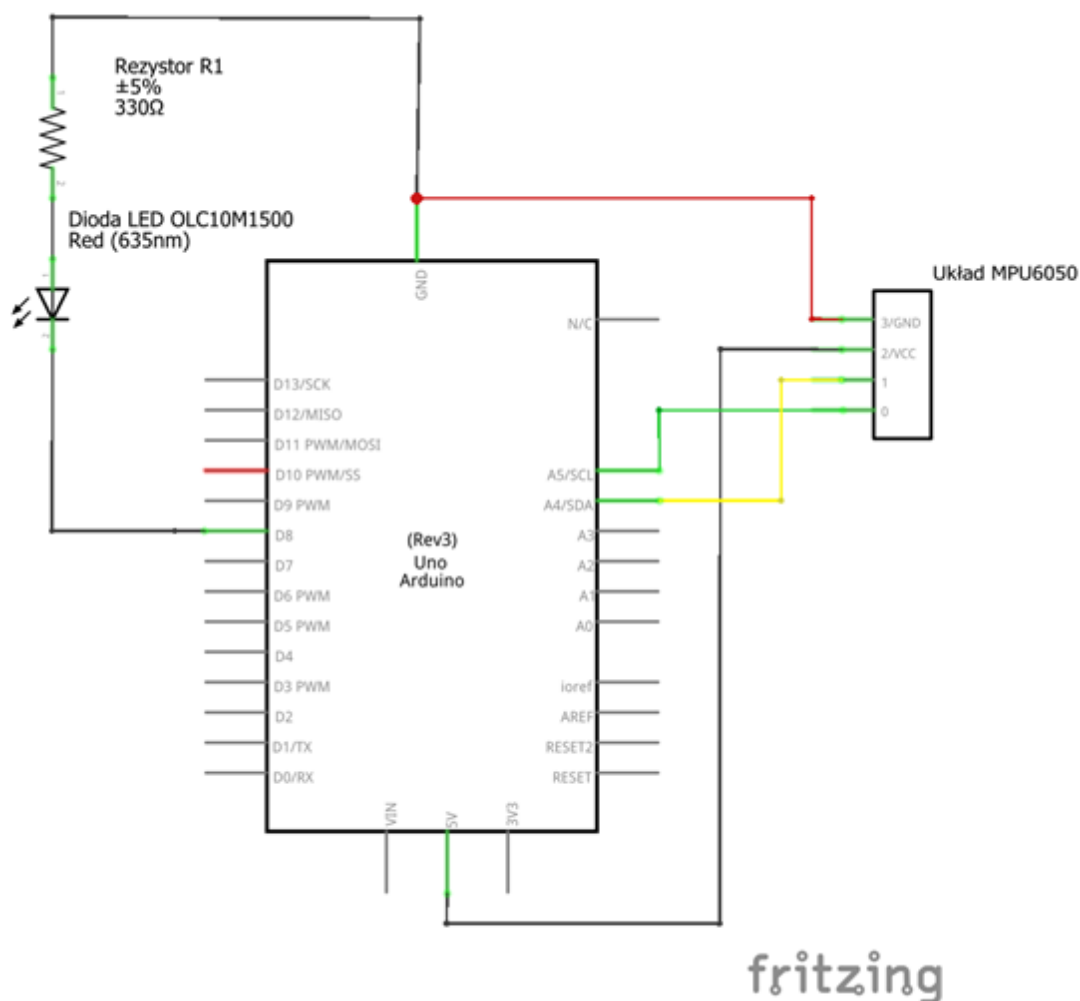
$n$  - ilość zwojów czynnych w sprężynie

$k$  - współczynnik sprężystości wyrażony w Newtonach na milimetr

Wybór odpowiedniej sprężyny był utrudniony przez ograniczoną dostępność sprężyn na rynku. Dlatego autor projektu zmuszony był dokonać pomiarów dostępnych sprężyn i obliczyć współczynnik sztywności sprężyny. Na podstawie tych kalkulacji projektodawca mógł dokonać wyboru sprężyny spełniającej powyższe kryteria. Do projektu inżynierskiego wybrano sprężynę nr 2. (Tab. 4.2), która jako jedyna, przy połączeniu równoległym z drugą taką samą sprężyną, spełniała kryterium dotyczące długości oraz współczynnika sprężystości. Sprężyny wykorzystano do połączenia pośredniczącego pomiędzy tarczą, a profilem o przekroju kwadratowym w układzie mechanicznym. Porównując dla przykładu wyniki symulacji przemieszczenia tarczy (Rys. 3.8) przy użyciu sprężyny nr 5. (Tab. 4.2) z parametrami z wynikami tej samej symulacji z użyciem sprężyny nr 2. (Rys. 3.7), można zauważyć, że zbyt duży współczynnik sprężystości nie pozwoli wystarczająco wprowadzić w ruch tarczy połączonej z sprężyną.

## **4.2. Układ elektroniczny**

Podpunkt 4.2 zawiera krótki opis układu elektronicznego, oraz opisany w podpunktach kod programu (którego zadaniem jest współpracować z płytką Arduino Uno przez wysyłanie i odbieranie danych do komputera) oraz omówienie połączenia układu elektronicznego z układem mechanicznym.



**Rys. 4.3. Schemat układu elektronicznego wykonany w programie Fritzing.**

Rys. 4.3 przedstawia połączenie układu MPU6050 funkcjonującego jako przyśpieszomierz oraz diodę LED będącą diodą sygnalizującą rozpoczęcie i zakończenie rejestracji czasu trafienia pocisku w tarczę. W celu zabezpieczenia diody przed spalaniem należało szeregowo połączyć ją z rezystorem R1 o wartości rezystancji 330 Ω. Układ MPU6050 podłączono według sposobu opisanego w dokumentacji technicznej. Sposób ten również opisany jest w podpunkcie 3.3 (nr 8).

#### **4.2.1. Kod programu Arduino**

Zamieszczono tutaj fragment zapisu kodu odpowiadającego za tworzenie tablicy dynamicznej wyników pomiaru z akcelerometru oraz za obsługę portu szeregowego połączonego z komputerem wraz z krótkim opisem głównych zadań wykonywanych przez kod.

```

1. void loop()
2. {
3.   int wysylane =1;
4.   if(wysylane==1);
5.   {
6.     Vector normAccel = mpu.readNormalizeAccel(); //tworzenie dynamicznej tablicy z wynikow pomiarowych
7.
8.     Serial.print(" XI = ");
9.     Serial.print(normAccel.XAxis); //znormalizowany pomiar z osi x
10.    Serial.print(" YYYI = ");
11.    Serial.print(normAccel.YAxis); //znormalizowany pomiar z osi y
12.    Serial.print(" ZZZI = ");
13.    Serial.println(normAccel.ZAxis); //znormalizowany pomiar z osi z
14.    delay(200);
15.  }
16.  if(Serial.available()) //aktywowanie portu szeregowego
17.  {
18.    int data = Serial.read(); //zdefiniowanie zmiennej, która będzie odbierana z portu szeregowego
19.    if(data == 'A')
20.    {
21.      digitalWrite(8, HIGH); //Włączenie diody
22.    }
23.    if(data == 'C')
24.    {
25.      digitalWrite(8, LOW); //Wyłączenie diody
26.    }
27.  }
28. }

```

**Rys. 4.4. Fragment kodu odpowiedzialny za działanie kodu w pętli Arduino Uno.**

Powyżej przedstawiono kod głównej pętli programu, do którego dodana została biblioteka MPU6050. Pętla ta realizuje kod odpowiadający za zapisywanie do portu szeregowego wyników pobieranych z tablicy dynamicznej akcelerometru co 100 ms. Kod jest odpowiedzialny za sprawdzenie aktywności portu szeregowego i za odbieranie wiadomości z komputera osobistego. Program w Microsoft Visual Studio został zrealizowany tak, aby wysyłał do płytki Arduino Uno wiadomość „A” informującą o włączeniu diody sygnalizującej gotowość do oddania strzału oraz wiadomość „C” informującą o wyłączeniu diody bezpośrednio po trafieniu pocisku w tarczę.

#### 4.2.2. Połączenie układu elektronicznego z mechanicznym

Po opracowaniu i zaprogramowaniu układu elektrycznego należało go połączyć z układem mechanicznym. Autor projektu umieścił akcelerometr pod tarczą w taki sposób, aby oś „z” czujnika była skierowana prostopadle w stronę profilu o przekroju kwadratowym, natomiast oś „y” skierowana była względem osi „z” prostopadle w górę. Takie rozwiązanie wykonane w ramach pracy aplikacji umożliwia między innymi obliczenie różnicy wartości przyspieszenia na wszystkich osiach bieżącego pomiaru odejmując go od poprzedniego pomiaru. Wynik równy lub większy 1 powoduje rejestrację momentu uderzenia pocisku w tarczę.

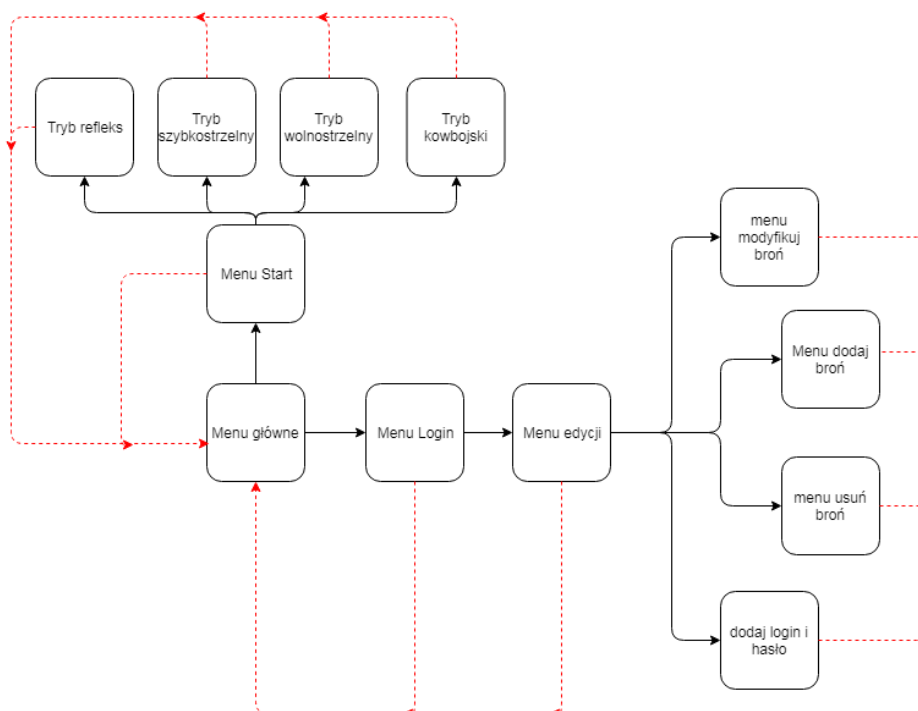


Rys. 4.5. Zdjęcie czujnika zamocowanego na tarczy.

Rys. 4.5 przedstawia widok rzeczywisty zamocowanego czujnika przyspieszenia znajdującego się bezpośrednio obok sprężyny. Kierunek osi widocznych na rysunku zapewnia prawidłowy odczyt wartości przyspieszenia z akcelerometru.

#### 4.3. *Interfejs programu*

Jest to ostatni punkt tego rozdziału, który dotyczy interfejsu programu. Zbudowany jest z części edycyjnej oraz z części realizującej tryb refleks. Rys. 4.6 przedstawia zależności połączonych między sobą interfejsami programu.

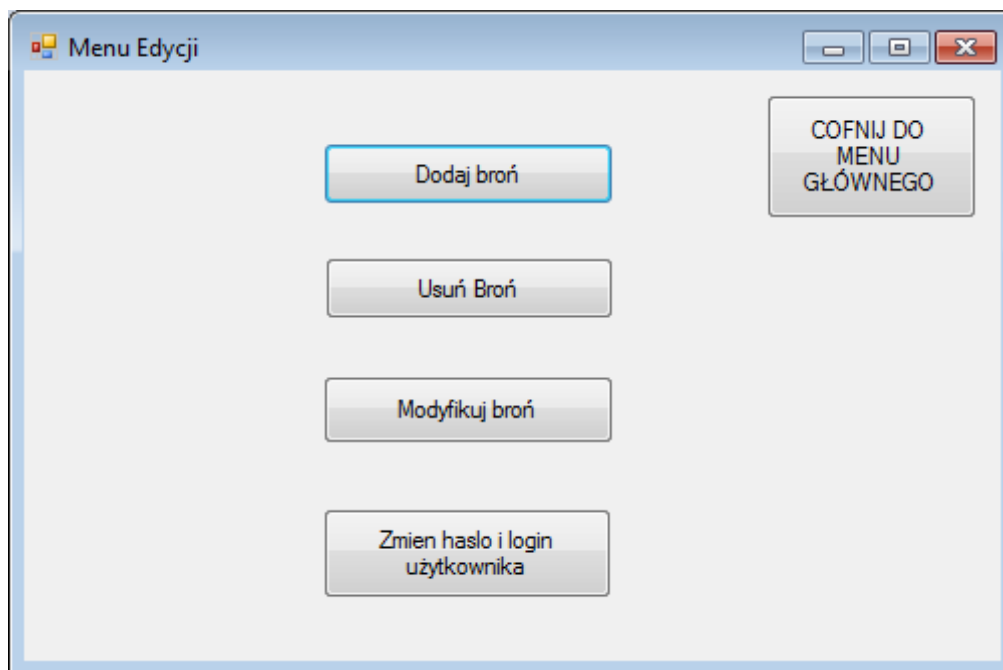


**Rys. 4.6. Graficzne przedstawienie zależności pomiędzy sesjami aplikacji.**

Start aplikacji zaczyna się w menu głównym programu, a następnie przy pomocy przycisków włącza kolejne interfejsy oprogramowania. Zależności dotyczące powrotu do menu głównego oznaczono przerywaną linią, a ich przycisk znajduje się, w każdej sesji, w prawym górnym rogu.

#### **4.3.1. Komunikacja z bazą danych**

Przedstawiona na Rys. 4.7 część interfejsu odpowiada za przeniesienie użytkownika do sesji związanych z dodawaniem/usuwaniem/modyfikacją bazy danych i interfejsu odpowiadającego za zmianę hasła. W tym punkcie opisano również połączenie się opracowanego w Microsoft Visual Studio programu do badania refleksu snajpera z bazą danych, wykorzystywaną w dalszych częściach opisu interfejsu.



**Rys. 4.7. Interfejs graficzny odpowiadający za menu edycji.**

Widoczne na przedstawionym powyżej interfejsie przyciski „dodaj broń”, „usuń broń” oraz „modyfikuj broń” uaktywniają określony interfejs związany z bazą danych. Opis ich działania znajduje się w poniższych podpunktach. Przycisk „Zmien hasło i login użytkownika” ma przypisany osobny punkt i odpowiada za zmianę hasła i loginu użytkownika wpisywanego w formularzu aplikacji „Login”.

```

1. SqlConnection conn = new SqlConnection(@"server=Komputer\Dazzyl; trusted_connection=false; database=ASG; User ID=sa;
2. Password=lolek1234");
3. // Wyżej wywołano połączenie do servera, a następnie bazy danych ASG znajdującym się na serverze
4. conn.Open(); // otwarcie połączenia z baza danych dla określonego wyżej adresu conn
5. SqlCommand sc = new SqlCommand("select * from bronie_ASG", conn); // tworzenie nowego zapytania i wysyłanie do bazy danych ASG
6. SqlDataReader reader = sc.ExecuteReader(); // odczytanie wszystkich elementów z bazy danych dotyczących zapytania do programu
7. conn.Close(); // zakończenie połączenia z baza danych

```

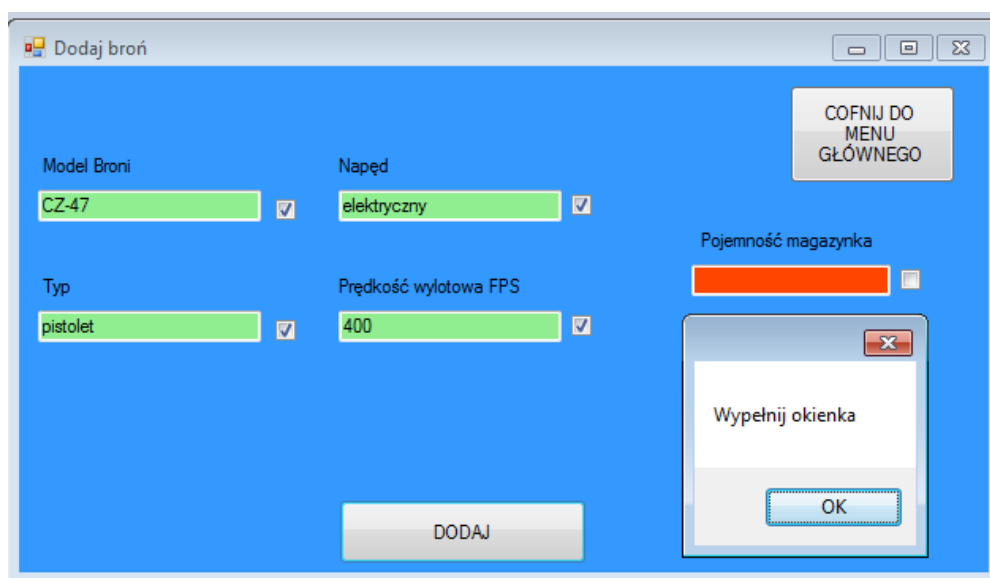
**Rys. 4.8. Kod odpowiadający za połączenie się z bazą danych.**

W zapisie kodu widocznym na Rys. 4.8 pokazany jest sposób połączenia i wysyłania zapytań do bazy danych z aplikacji opracowanej w języku C# (opis działania znajduje się w komentarzach na Rys. 4.1). Kod ten spełnia ważną rolę w funkcjonowaniu układu do badania refleksu snajpera. Wykorzystywany jest między innymi podczas magazynowania i odczytywania danych związanych z wartościami czasowymi strzelca i parametrami broni. Poniżej opisano w podpunktach główne zadania związanych z obsługą bazy danych.



### a) Menu „Dodaj broń”

Poniższy formularz aplikacji Windows „dodaj broń” odpowiada za połączenie się z bazą danych ASG i wpisanie do tabeli broni ASG parametrów odnoszących się do nowej broni, której wartość prędkości wylotowej mieści się w zakresie od 400 FPS (121 m/s) do 500 FPS (152 m/s).

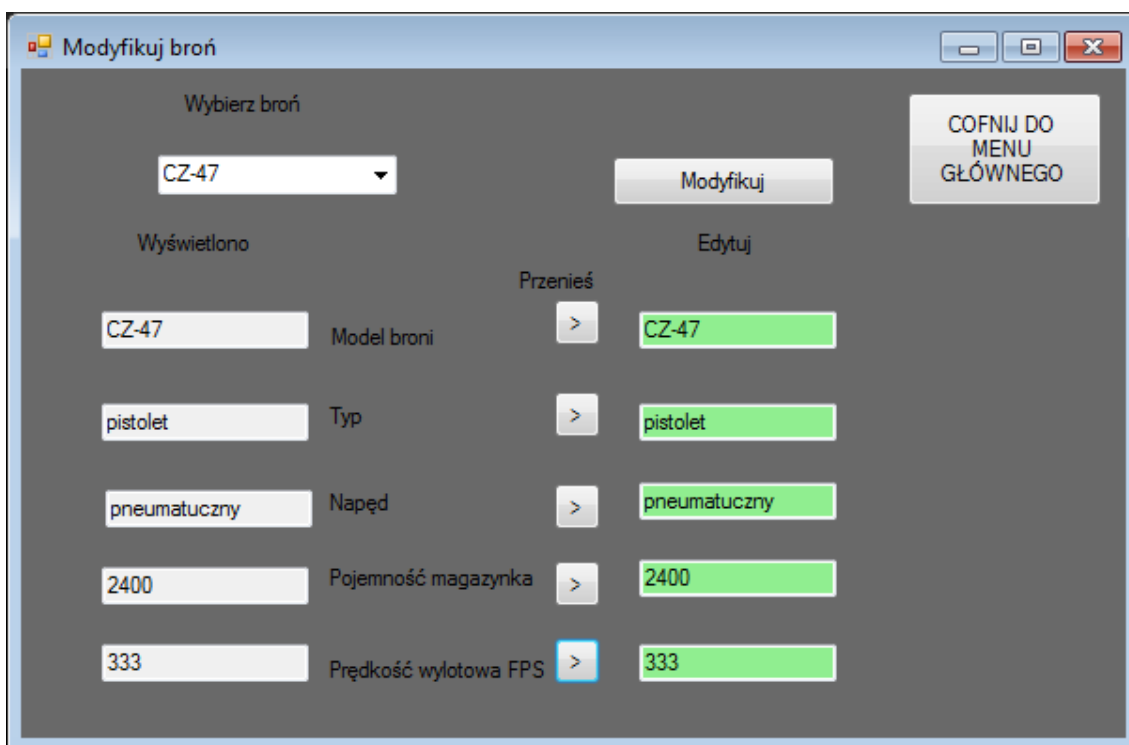


Rys. 4.9. Interfejs graficzny części programu odpowiadającej za dodawanie broni.

Na Rys. 4.9 widać komunikat „wypełnij okienka”. Odnosi się on do wpisywania wartości do pól tekstowych. Jeśli jedno z nich nie zostało uzupełnione to „okienko” danego pola tekstowego podświetla się na czerwono, Jednocześnie ukazuje się komunikat „wypełnij okienka”. Jeśli „okienko” było wypełnione, podświetla się na zielono i pojawia się komunikat „Dodano broń”. Dodatkowo zastosowano zabezpieczenie blokujące możliwość wpisania liczby lub numeru do nieodpowiednich pól tekstowych (na przykład prędkość wylotową można wyrazić tylko za pomocą liczb).

### b) Menu „Modyfikuj broń”

Przedstawiony formularz aplikacji Windows „dodaj broń” odpowiada za połączenie z bazą danych ASG i wysyłanie do niej zapytań modyfikujących parametry broni znajdujące się w tabeli „bronie\_ASG”.

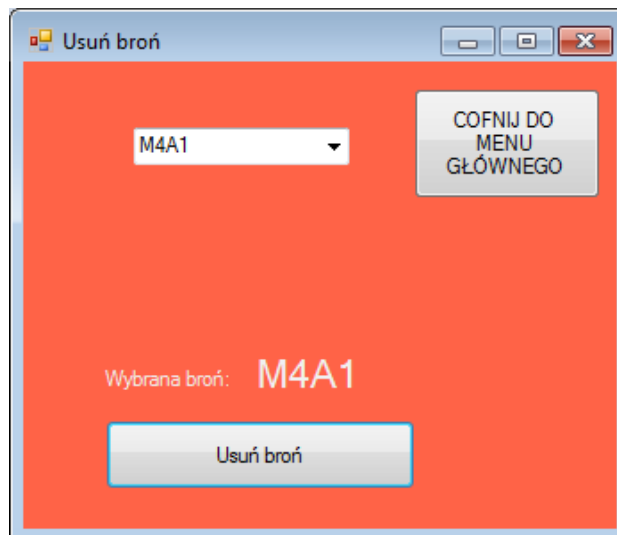


**Rys. 4.10.** Interfejs graficzny części programu odpowiadającej za modyfikację broni.

W polu wyboru (pod napisem „wybierz broń”) określono, jaką broń z tabeli „bronie\_ASG” użytkownik chce zmodyfikować, wyświetlając po lewej stronie jej parametry. Przyciski oznaczony „>” odpowiada za przeniesienie z parametru pola tekstowego z lewej na prawą stronę. Po naciśnięciu przycisku „modyfikuj” program najpierw usuwa z tabeli „bronie\_ASG” wybraną wcześniej w polu wyboru broń, a następnie dodaje nową broń z właściwymi jej parametrami. W tym formularzu aplikacji również skorzystano z zabezpieczeń dotyczących uzupełniania pól tekstowych, jak w formularzu aplikacji „dodaj broń”.

### c) Menu „Usuń broń”

Formularz aplikacji Windows „usuń broń” odpowiada za połączenie z bazą danych ASG i wysłanie do niej zapytania o usunięcie broni z tabeli „bronie\_ASG” zaznaczoną w polu wyboru, która znajduje się w tej tabeli.

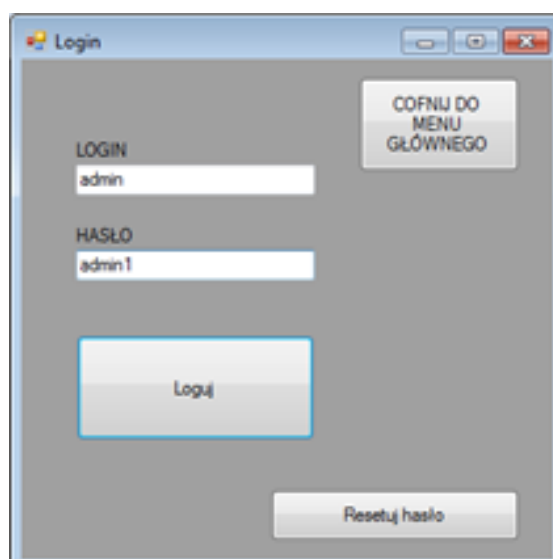


**Rys. 4.11.** Interfejs graficzny części programu odpowiadającej za usuwanie broni.

Na powyższym rysunku przedstawiono przykładowo wybraną broń „M4A1” zaznaczoną w polu wyboru. Po wciśnięciu przycisku „usuń broń” zostanie wysłane do bazy danych ASG zapytanie o treści "delete from bronie\_ASG where model\_broni=M4A1".

#### **4.3.2. Zabezpieczenia programu**

Podpunkt ten zawiera opis interfejsu odpowiadającego za zabezpieczenie, którego zadaniem jest ograniczanie dostępu do modyfikacji bazy danych przez nieuprawnionych użytkowników oraz edytowanie przez nich hasła i loginu.



**Rys. 4.12.** Interfejs graficzny aplikacji formularzu Windows „Login”.

Na Rys. 4.12 w pole tekstowe „LOGIN”, należy wpisać ustalony login, a w pole tekstowe „HASŁO”, należy wpisać hasło wybrane przez użytkownika. Następnie trzeba wcisnąć przycisk „Loguj”. Jeżeli wpisane hasło i/lub login są niepoprawne pojawia się komunikat "Error: Źle wpisane hasło lub login". Jeżeli natomiast dane wpisano poprawnie otwiera się „menu edycji”.

```
1. private void reset_Click(object sender, EventArgs e)
2.     {
3.         Properties.Settings.Default.Username = "admin"; //ustanowienie dostępu i przypisanie mu nowej wartosci uzytkownika
4.
5.         Properties.Settings.Default.Password = "admin1"; // ustanowienie dostępu i przypisanie mu nowej wartosci hasla
6.         Properties.Settings.Default.Save(); // utrwalenie zmian miedzy formularzami aplikacji metoda Save
7.         MessageBox.Show("Haslo zresetowane do ustawien domyslonych"); //komunikat o zresetowaniu ustawien
8.     }
```

**Rys. 4.13. Fragment kodu resetujący hasło i login dla formularzy aplikacji „Login”.**

Przedstawiony kod odpowiada za resetowanie hasła i loginu, w formularzu „Login”, przyciskiem „Resetuj hasło”. Po użyciu opcji „Resetuj hasło” automatycznie generuje się login „admin” oraz hasło „admin1”(sposób działania opisany w komentarzach na Rys. 4.13).

**Rys. 4.14. Interfejs graficzny aplikacji formularzu Windows „Dodaj\_login\_haslo”.**

Interfejs „Dodaj\_login\_haslo” odpowiada za modyfikację bieżącego hasła i loginu na te wpisane w określonych polach tekstowe. W celu zatwierdzenia zmiany należy wcisnąć przycisk „Dodaj użytkownika”.

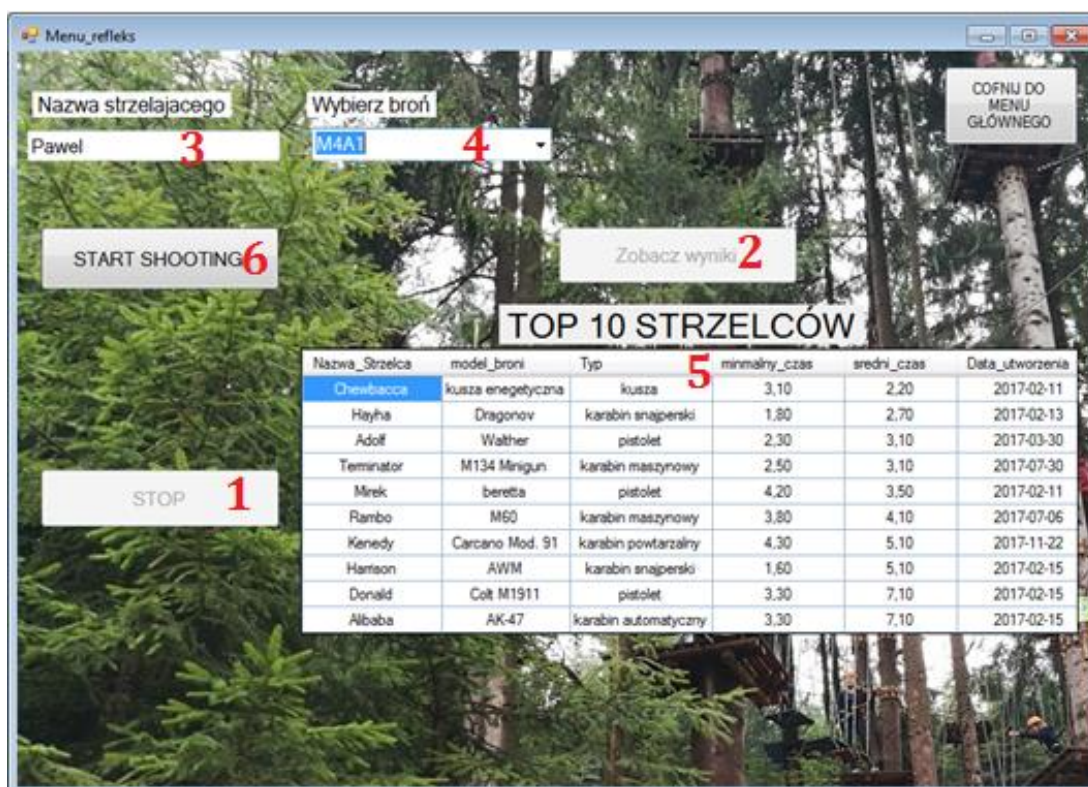
### 4.3.3. Główny Interfejs tryb refleks

Rys. 4.15 przedstawia interfejs programu „start” wraz z jego zadaniami. Opisano również interfejs „Menu refleks” i jego najważniejsze funkcje dotyczące badania refleksu snajpera strzelającego z broni typu ASG.



**Rys. 4.15. Interfejs graficzny aplikacji formularzu Windows „Dodaj\_login\_hasło”.**

Sesja „Start” odpowiada za wybranie trybu, z którego może skorzystać użytkownik. Poprzez kliknięcie przycisku odpowiedniego trybu włączą interfejs związany z wybranym trybem. Tryby „Wolnostrzelny”, „Szybkostrzelny” oraz „Kowbojski” to opcje przewidziane do realizacji w przyszłości. Niestety z powodu braku środków finansowych nie zrealizowano ich w tym projekcie, gdyż wymagały one zastosowania 2 dodatkowych stanowisk (jak na Rys. 5.1).



**Rys. 4.16.** Interfejs graficzny aplikacji formularza Windows „Menu refleks” odpowiadający za tryb refleks.

**Nr 1.** Przycisk „STOP” odpowiada za zatrzymanie badania refleksu strzelca w dowolnym momencie przy aktywnym przycisku „START SHOOTING”.

**Nr 2.** Przycisk „Zobacz Wynik” uaktywnia się po zakończeniu rejestracji refleksu strzelca.

**Nr 3.** Pole tekstowe, w które wpisana została nazwa strzelca. Jest ono potrzebne do późniejszej identyfikacji gracza w „Top strzelców” (jeśli zostanie zakwalifikowany) oraz podczas generowania pliku do formatu .pdf do odczytu. W przypadku nie wpisania w pole tekstowe nazwy strzelca, automatycznie generuje się nazwa strzelca jako „Stormtrooper”.

**Nr 4.** Pole wyboru, w którym użytkownik dokonuje decyzji dotyczącej modelu broni, z której zamierza strzelać. Nazwa wybranej broni jest uwzględniana generowaniu do pliku .pdf.

**Nr 5.** Dane wyświetlane są w siatce „Top strzelców” za pomocą wysyłanego do tabeli „Walhalla” zapytania o wartościach "select Nazwa\_Strzelca, model\_broni, Typ, minimalny\_czas, sredni\_czas, Data\_utworzenia from Walhalla order by sred-

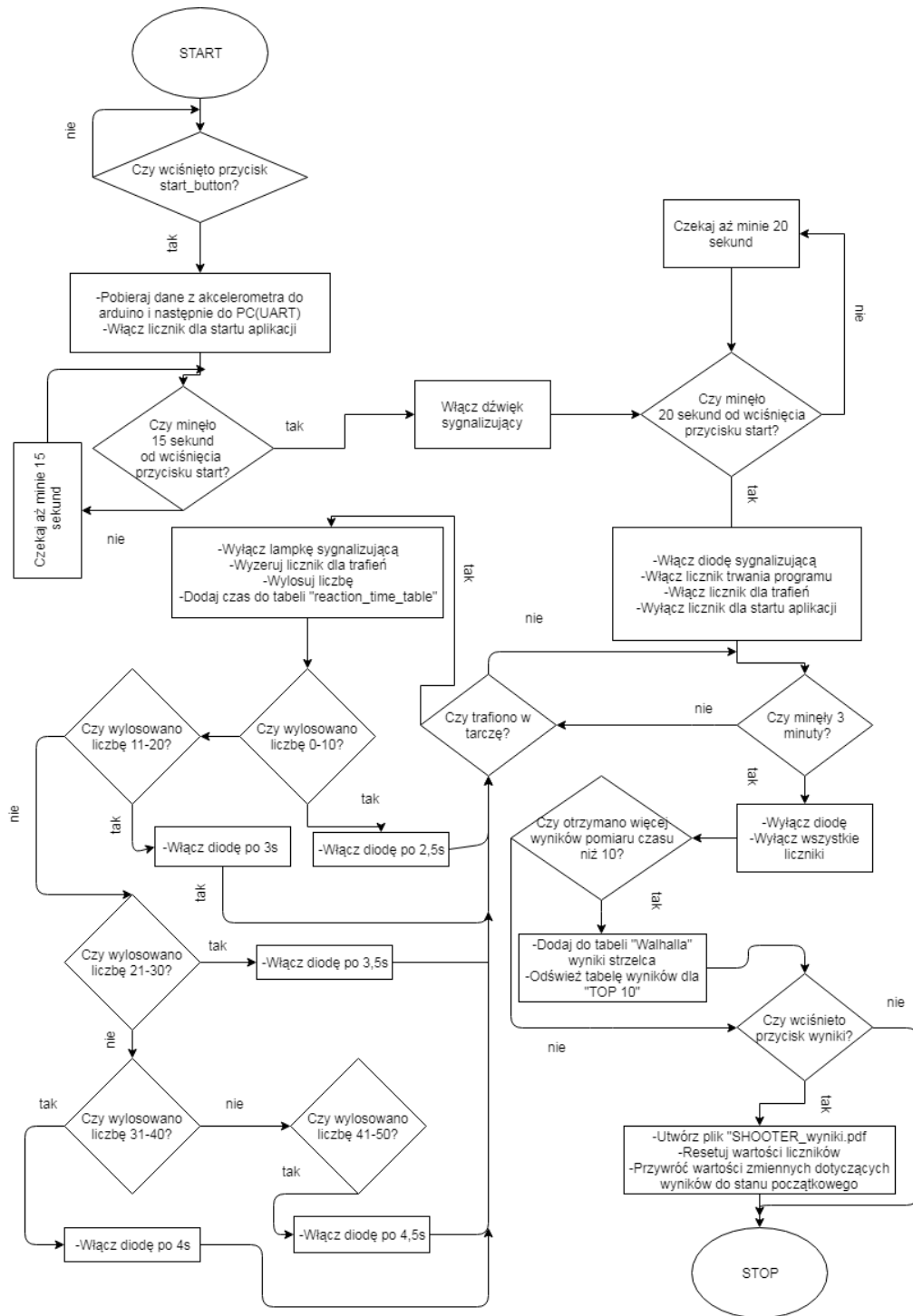
ni\_czas". Poprzez zablokowanie opcji przewijania wyników pokazywana jest tylko lista „TOP 10 strzelców”.

**Nr 6.** Przycisku „START SHOOTING” odpowiadająca za włączenie się programu do badania refleksu strzelca. Zasada działania ww. przycisku została ukazana w postaci algorytmu przedstawionego na Rys. 4.18.

```
1. //_____FUNKCJA2 final_AVG_____
2.     private void final_AVG() //liczenie sredniej strzelca w domysle jako refleks wszystkich ;oczn z tabeli
3.     {
4.         if (i_position_avg < 300) //wartosc poczatkowa i_position_avg = 0 , petla odpowiada za przesuwanie miejsc w tabelii
5.         {
6.             double min_liczb_avg = Convert.ToInt32(reaction_time_table[i_position_avg]);
7.             // powyza linijka odpowiada za przypisywanie do zmiennej min_liczb_avg wartosci czasu z tabeli reaction_time_table
8.             i_position_avg++; //inkremetacja pozycji w tabeli
9.
10.
11.             if (min_liczb_avg > 0) // wybieranie z tabeli wartosci czasow wiekszych od 0
12.             {
13.                 dzielnik++; // liczba dokonanych pomiarow
14.                 nr_first_Avg = nr_first_Avg + min_liczb_avg; // tworzenie czasu ogolnego w mikrosekundach
15.             }
16.             if (i_position_avg == 300) // jesli pozycja w tabeli jest rowna 300 to wykonaj kod ponizej
17.             {
18.                 AvgValue = (nr_first_Avg / 100) / dzielnik; // tworzenie sredniej i zamiana mikrosekund na milisekundy
19.             }
20.         }
21.     }
```

**Rys. 4.17.** Fragment kodu odpowiadającego za liczenie średniej.

Funkcja „Final\_AVG” jest najważniejszą funkcją programu do badania refleksu strzelca. Jej zadaniem jest wygenerowanie zmiennej globalnej typu „double” o nazwie „AvgValue” jako wartości średniego czasu strzelca, który jest wartością refleksu snajpera. Na powyższym rysunku została opisana obróbka danych podczas liczenia średniej.



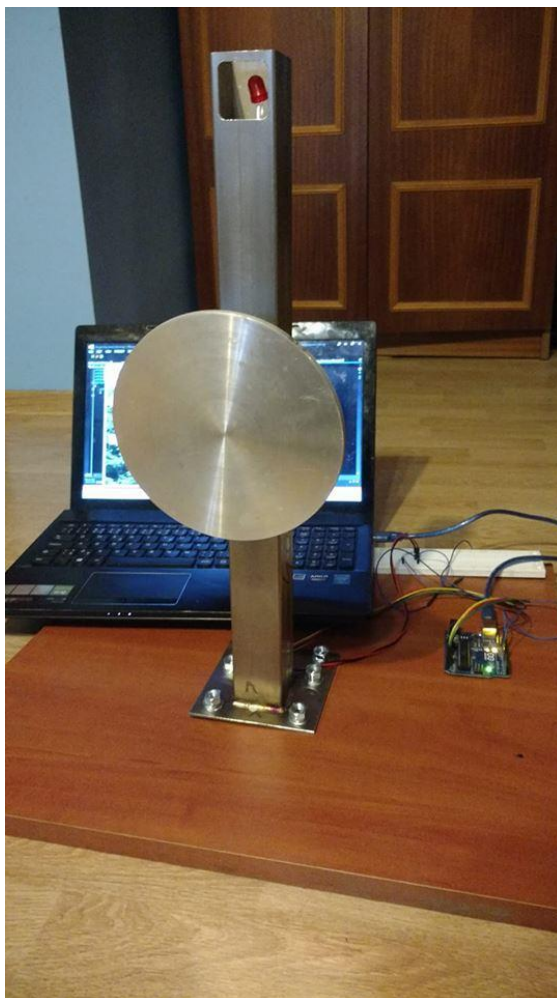
**Rys. 4.18. Algorytm działania programu po wciśnięciu przycisku START SHOOTING.**

Przedstawiony algorytm odpowiada za proces odczytywania wartości czasowych określanych mianem refleksu strzelca. W powyższym algorytmie nie uwzględniono przycisku „STOP”, który jest opisany jako Nr.1 pod Rys. 4.14.



## 5. Testy, BHP, przyszły rozwój

W rozdziale 5 autor projektu opisał wyniki przeprowadzonych testów aplikacji. Przeprowadzenie tych testów było konieczne do sprawdzenia, czy aplikacja może być wykorzystana w projekcie inżynierskim. Przeprowadzenie gruntownych testów pozwoliło na sprawdzenie nieprzewidzianych błędów układu. Następny podpunkt dotyczy zasad bezpieczeństwa korzystania ze stanowiska, aby badanie refleksu gracza przecho-  
dziło prawidłowo i nie doprowadzało do urazów czy kontuzji. W ostatnim punkcie autor opisuje koncepcję przyszłego rozwoju aplikacji.



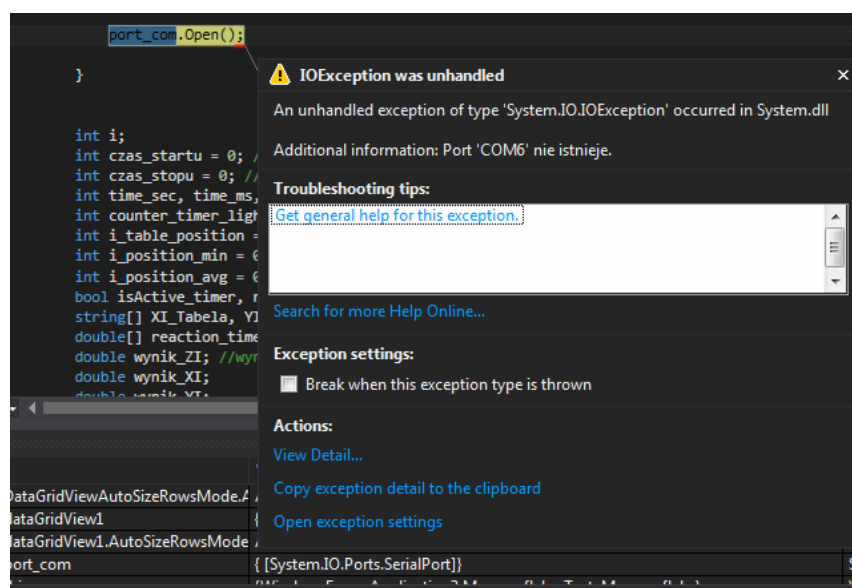
**Rys. 5.1. Stanowisko do badania refleksu snajpera.**

Na Rys. 5.1 widać stanowisko, przy użyciu którego przeprowadzano testy aplikacji. Stanowisko składa się z 3 połączonych ze sobą układów: mechanicznego,

elektronicznego oraz programowego. Testy były przeprowadzone głównie w warunkach polowych w celu sprawdzenia czy zmienne tj. niska temperatura nie mniejsza niż 2° C oraz prędkość wiatru nie większą niż 10 km/h będą miały wpływ na funkcjonowanie programu do badania refleksu snajpera.

### 5.1. Testy aplikacji

Testy aplikacji wykonano bez użycia specjalistycznych narzędzi. Oprogramowanie posiadało własne zabezpieczenia programowe umożliwiające uniknięcie znacznej części błędów. Czynniki takie jak prędkość wiatru i niska temperatura nie miały wpływu na funkcjonowanie programu. Odłączenia kabla USB, łączącego komputer osobisty z płytką Arduino Uno, przy otwartym jednocześnie interfejsie „Menu refleks” spowodowało wystąpienie błędu. Niestety nie udało się naprawić zaistniałego problemu, lecz nie był on błędem krytycznym powodującym znaczne uszkodzenia elementów bazy danych programu do badania refleksu.



Rys. 5.1. Okno błędu podczas działania trybu refleks.

Komunikat tego błędu widoczny jest na rysunku 5.1. Błąd występuje, tylko podczas korzystania z interfejsu „Tryb refleks” i równoczesnego odłączenia przewodu USB od układu Arduino (lub braku podłączenia przewodu).

Dlatego podczas użytkowania należy uważać, aby omyłkowo nie odłączyć przewodu USB od układu Arduino.

## **5.2. BHP**

Na prawidłowy przebieg badania wpływa wiele aspektów zarówno dotyczących układu elektronicznego i mechanicznego, jak i czynności wykonywanych przez strzelca. Szeregiem czynności, jakie musi wykonać strzelec zarówno przed, w trakcie jak i po badaniu są czynności związane z organizacją stanowiska pracy (BHP). W podpunkcie 5.2. zostaną pokrótce przedstawione i opisane najważniejsze zasady BHP, jakich musi przestrzegać strzelec, nie tylko aby badanie mogło przebiec prawidłowo, ale również by przebiegało bezpiecznie.

- **Obowiązujące środki ostrożności stosowane przed rozpoczęciem badania refleksu strzelca:**

1. Stanowisko do badania refleksu strzelca powinno być ustawione na płaskiej powierzchni.
2. Stanowisko do badania refleksu strzelca powinno być ogrodzone materiałem zabezpieczającym w celu uniknięcia niebezpieczeństwa postrzału osób postronnych.
3. Strzelec ma obowiązek posiadać ubrane okulary ochronne (zabezpieczające oczy strzelca) oraz maskę ochronną (zabezpieczającą część twarzy poniżej poziomu oczu).
4. Strzelec ma obowiązek upewnić się o prawidłowym funkcjonowaniu broni przed oddaniem strzału oraz niezwłocznie poinformować instruktora o ewentualnych nieprawidłowościach.

- **Obowiązujące środki ostrożności stosowane w trakcie badania refleksu strzelca:**

1. Przy stanowisku do badania refleksu strzelca może znajdować się wyłącznie instruktor nadzorujący badanie oraz strzelec.
2. Strzelec może oddać strzał tylko w obecności instruktora.
3. Strzelcowi wolno oddać strzał tylko, gdy znajduje się w odległości 10 metrów od tarczy układu badającego refleks strzelca.
4. Wszelkich czynności związanych z obsługą broni dokonuje się wyłącznie z lufą skierowaną w kierunku tarczy.

- **Obowiązujące środki ostrożności stosowane po zakończeniu badania refleksu strzelca:**

1. Strzelec zobowiązany jest rozładować (wyjmując z niej magazynek) oraz niezwłocznie zdać ją instruktorowi nadzorującemu badanie.

2. Instruktor nadzorujący badanie ma obowiązek zabezpieczyć broń i umieścić ją w bezpiecznym miejscu przeznaczonym do jej przechowania, tak aby broń była skierowana lufa w dół.

### **5.3. *Rozwój aplikacji w przyszłości***

Oprogramowanie do badania refleksu snajpera strzelającego z broni typu ASG posiada wiele możliwości rozwoju. Ponadto może zostać rozszerzony o tryby związane z badaniem celności strzelca, co może stać się tematem do kolejnych badań. Poniżej przedstawiono przykładowe koncepcje rozwoju układu:

- Optymalizacji kodu – poprawa funkcjonalności programu, podczas dodawania nowych trybów.
- Dodanie trybów sprawdzania celności – wyposażenie układu o dodatkowe stanowiska do badania celności strzałów oddawanych przez strzelca oraz odpowiednia modyfikacja programowa umożliwiająca ich wykorzystanie.
- Bazy danych o rozszerzeniach .xlc lub .csv - nie wymagające konfiguracji MySQL (wykorzystujące program Microsoft Excel).
- Przechowywanie bazy danych na stronie internetowej – zabezpieczenie przed utratą bazy danych z komputera.
- Roczny wykres wyników – ukazanie w spisie miesięcznym w przekroju całorocznym, najlepszych wyników dla danej broni, częstotliwości miesięcznej korzystania z danego trybu oraz możliwość czyszczenia bazy danych po upływie roku.

## 6. Podsumowanie i wnioski

Inspiracją dla powstania projektu (Układu do badania refleksu snajpera strzelającego z broni ASG) stała się pasja i zamiłowanie do strzelania z broni typu ASG. Stąd cała koncepcja projektu złożona z założeń, znajdujących się w rozdziale 2, mających na celu opracowanie układu precyzyjnie badającego refleks snajpera. Autor projektu zdecydował się na niego nie tylko ze względu na jego innowacyjność czy efektywność, ale przede wszystkim na jego użyteczność w świecie pasjonatów ASG. Założenia postawione przez pomysłodawcę projektu, na początku pracy inżynierskiej, stały się możliwe do zrealizowania przy wykorzystaniu najlepiej dobranych elementów układu, począwszy od języka programowania, przez układ elektroniczny aż po materiały, z których wykonano układ mechaniczny. Dla wykonawcy projektu ważne jest, aby napisana aplikacja cechowała się przejrzystością oraz łatwością w obsłudze. Dlatego wybranym językiem programowania był język C#, który jest przeznaczony dla systemu Windows, co umożliwiło powszechne korzystanie z aplikacji na komputerze osobistym. Ponadto wykorzystanie bibliotek .NET tj. „Media”, „Data.SqlClient” czy IO umożliwiło wzbogacenie aplikacji np. biblioteka „Media” odpowiedzialna jest za odtwarzanie dźwięku sygnalizującego rozpoczęcie lub zakończenie procesu badania refleksu snajpera. Układ elektroniczny został wykonany na płytce Arduino Uno, co bardzo usprawnia całe funkcjonowanie układu i zapewnia mu stabilność. Wielokrotne symulacje dotyczące układu mechanicznego, pozwoliły stwierdzić czy zmiana długości sprężyny jest dostosowana do prędkości wylotowej pocisku. Wszystkie wyżej wymienione cechy projektu inżynierskiego, pozwalają na rozwój aplikacji w przyszłości.

## 7. Bibliografia

- [1] Robert Temple.: Geniusz Chin: 3000 lat nauki, odkryć i wynalazków, wydawnictwo Ars Polona, Warszawa 1994r.
- [2] Albahari B. Albahari S.: C# 6.0 w pigułce, wydawnictwo Helion, Rybnik 2018
- [3] Wprowadzenie do języka C#  
[https://msdn.microsoft.com/pl-pl/library/z1zx9t92\(v=vs.120\).aspx](https://msdn.microsoft.com/pl-pl/library/z1zx9t92(v=vs.120).aspx)  
dostęp: 18.02.2018
- [4] Dokumentacja biblioteki iTextSharp  
<https://sourceforge.net/projects/itextsharp/>,  
dostęp 18.02.2018
- [5] Dokumentacja biblioteki Media  
[https://msdn.microsoft.com/en-us/library/system.media\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.media(v=vs.110).aspx)  
dostęp 18.02.2018
- [6] Dokumentacja biblioteki Data.SqlClient  
[https://msdn.microsoft.com/pl-pl/library/system.data.sqlclient\(v=vs.110\).aspx](https://msdn.microsoft.com/pl-pl/library/system.data.sqlclient(v=vs.110).aspx)  
dostęp 18.02.2018
- [7] Dokumentacja biblioteki IO  
[https://msdn.microsoft.com/pl-pl/library/system.io\(v=vs.110\).aspx](https://msdn.microsoft.com/pl-pl/library/system.io(v=vs.110).aspx)  
dostęp 18.02.2018
- [8] Dokumentacja MySQL:  
<https://dev.mysql.com/doc>  
dostęp: 19.02.2018
- [9] Wprowadzenie do Arduino  
<https://www.arduino.cc/en/Guide/Introduction>  
dostęp 18.02.2018
- [10] Arduino Środowisko jak zacząć programować  
<https://forbot.pl/blog/kurs-arduino-srodowisko-jak-zaczac-programowac-id936>  
dostęp 18.02.2018

- [11] Dokumentacja biblioteki MPU6050  
[www.jarzebski.pl/arduino/czujniki-i-sensory/3-osiowy-zyroskop-i-akcelerometr-mpu6050.html](http://www.jarzebski.pl/arduino/czujniki-i-sensory/3-osiowy-zyroskop-i-akcelerometr-mpu6050.html)  
dostęp 18.02.2018
- [12] Lis M.: Praktyczny kurs C# , wydawnictwo Helion, Rybnik 2018
- [13] Wprowadzenie do programu Visual Studio:  
<https://msdn.microsoft.com/pl-pl/library/ms165079.aspx>, dostęp: 15.02.2018
- [14] Słownikowa definicja czym jest CAD  
<https://www.techopedia.com/definition/2063/computer-aided-design-cad>  
dostęp 18.02.2018
- [15] Autodesk Inventor Professional  
<https://www.autodesk.com/education/free-software/inventor-professional>  
dostęp 18.02.2018
- [16] Poradnik zasada działania akcelometru  
<http://roboblog.eu/2016/04/07/arduino-poradnik-8-zyroskopakcelometr-mpu-6050/>  
dostęp 19.02.2018