



POLSKO-JAPOŃSKA AKADEMIA  
TECHNIK KOMPUTEROWYCH

**WYDZIAŁ INFORMATYKI**  
**KATEDRA METOD PROGRAMOWANIA**  
**PROGRAMOWANIE APLIKACJI BIZNESOWYCH**

**Kamil Kacprzak**  
s14004

**Zastosowanie technologii rzeczywistości rozszerzonej w biznesie, ze szczególnym uwzględnieniem rękawicy-kontrolera**

Praca inżynierska  
dr Krzysztof Bartczko

**Warszawa, Czerwiec, 2020**



# **Spis treści**

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Zagadnienia</b>	<b>3</b>
2.1	Historia kreowania rzeczywistości . . . . .	3
2.2	Wpływ kultury na rozwój technologii . . . . .	5
2.2.1	Kinematografia . . . . .	5
2.2.2	Marzenia senne . . . . .	7
2.3	Rodzaje rzeczywistości . . . . .	10
2.3.1	XR . . . . .	10
2.3.2	VR . . . . .	11
2.3.3	AR . . . . .	12
2.3.4	MR . . . . .	13
2.4	Potencjał technologii . . . . .	14
<b>3</b>	<b>Prezentacja kreowanej rzeczywistości</b>	<b>15</b>
3.1	Okulary VR . . . . .	15
3.2	Smart-fon jako urządzenie do prezentacji . . . . .	15
3.3	Kontrolery i IoT . . . . .	15
3.3.1	Problemy kontrolerów . . . . .	15
3.3.2	Działanie czujników . . . . .	15
3.4	Dłoń jako kontroler . . . . .	15
<b>4</b>	<b>Rękawice jako kontrolery</b>	<b>16</b>
4.1	Rozwój rękawicy jako kontrolera . . . . .	16
4.2	Podział rękawic-kontrolerów . . . . .	18
4.2.1	Klasyczne . . . . .	18
4.2.2	Nasadowe . . . . .	22
4.2.3	Egzoszkielety . . . . .	24

---

<b>5</b>	<b>Zastosowania w biznesie</b>	<b>25</b>
5.1	Lotnictwo i symulatory . . . . .	25
5.2	Medycyna . . . . .	25
5.3	Architektura . . . . .	25
5.4	Rozrywka . . . . .	25
<b>6</b>	<b>Projekt rękawicy</b>	<b>26</b>
6.1	Przegląd podzespołów użytych w projekcie . . . . .	26
6.1.1	Mikrokontroler . . . . .	27
6.1.2	Czujnik wygięcia . . . . .	29
6.1.3	Rezystor . . . . .	31
6.2	Budowa rękawicy . . . . .	33
6.3	Oprogramowanie mikrokontrolera . . . . .	37
6.4	Dalszy rozwój . . . . .	46
6.4.1	Problemy mikroprocesora . . . . .	46
6.4.2	Problemy budowy i czujników wygięcia . . . . .	47
6.4.3	Animacja modelu . . . . .	48
6.4.4	Błąd rotacji . . . . .	49
6.4.5	Problem obliczania przesunięcia . . . . .	50
6.5	Podsumowanie projektu . . . . .	50
<b>7</b>	<b>Podsumowanie</b>	<b>52</b>



# Spis rysunków

2.1	Interfejs aplikacji <i>Pokemon Go</i> wykorzystujący AR. . . . .	13
4.1	Pierwsza rękawica VR - Sayra Glove . . . . .	17
4.2	Przykładowe klasyczne rękawice VR. . . . .	21
4.3	Urządzenia nasadowe. . . . .	24
6.1	Opis wejść oraz wyjść Arduino Nano 33 BLE . . . . .	28
6.2	Sensor wygięcia własnego wykonania . . . . .	31
6.3	Oznaczenia rezystorów . . . . .	32
6.4	Układ dzielnika napięcia . . . . .	33
6.5	Poglądowy układ kontrolera przedstawiający sposób podłączenia dzielnika napięcia . . . . .	34
6.6	Efekt końcowy rękawicy-kontrolera . . . . .	35

## **Streszczenie**

## **Abstract**

# **Rozdział 1**

## **Wstęp**

Zjawisko tworzenia sztucznej rzeczywistości jest znane ludziom od wielu dekad. Pierwsze prace w tym obszarze zostały podjęte jeszcze przed powstaniem grafiki komputerowej. Sam koncept wykorzystania w tej technologii dloni jako kontrolera, powstał nie wiele później. W poniższych rozdziałach zostanie przedstawione jak technologia ta się rozwinęła od tamtej pory, oraz jak liderzy rynku rozwiązują problemy które są im stawiane przez ograniczenia współczesnej technologii. Tematem pracy jest “Zastosowanie technologii rzeczywistości rozszerzonej w biznesie, ze szczególnym uwzględnieniem rękawicy-kontrolera”. Przedstawiana praca pokazuje technologię rzeczywistości rozszerzonej oraz sposób w jaki jest ona prezentowana użytkownikom. Różne metody i podejścia sprawiają że rynek jest różnorodny oraz wypełniony wieloma rodzajami urządzeń które wspomagają realizm prezentowanych światów. Rozwój realizmu oraz wygody korzystania z tego rodzaju urządzeń sprawił że wiele firm coraz częściej wykorzystuje część tej technologii w celu optymalizacji procesów oraz kosztów w biznesie. Temat ten jest szerokim zagadnieniem z powodu wielu rodzajów rzeczywistości jakie są wykorzystywane. W poniższej pracy zostaną opisane elementy składające się na termin rzeczywistości rozszerzonej oraz w jakiej dziedzinie poszczególne jej obszary mają przewagę nad innymi typami. Praca skupia się w szczególności

---

na zastosowaniu technologii rękawic-kontrolerów w połączeniu z technologiami świata wirtualnego. Urządzenia te pozwalają na komunikację pomiędzy użytkownikiem a prezentowanymi światami w sposób realny oraz intuicyjny. W szczególności intuicyjność tego rozwiązania jest jednym z głównych powodów powstania tej pracy. Warto również zauważyć że rynek ten jest rynkiem rozwijającym się co zostanie pokazane na przykładzie rozwiązań już istniejących oraz skali zainteresowania tymi produktami. Produkty te zostaną szczegółowo opisane, ich podział ze względu na budowę oraz zastosowania które się różnią w zależności od implementowanych koncepcji. Praca ma na celu przeanalizować podejścia na jakie twórcy tych produktów się zdecydowali oraz wady i zalety technologii z której korzystają. Głównym wyzwaniem w tym obszarze jest rozwiązanie problemów nawigacji inercyjnej, czyli rodzaju nawigacji pozwalającego na określenie położenia oraz orientacji w systemie, zazwyczaj bez układu odniesienia. Temat ten nie posiada uniwersalnego rozwiązania w związku z czym dodatkowe elementy są często wykorzystywane w celu usprawnienia funkcjonowania systemu. Oprócz informacji dotyczących położenia bardzo ważnym aspektem jest możliwość śledzenia palców dłoni. W ten sposób są spełnione podstawowe wymagania kontrolera, jednak aby wyróżnić się na rynku wiele firm prezentuje dodatkowe rozszerzenia wzmacniające odczucia podczas użytkowania, takie jak wibracje podczas kolizji w świecie wirtualnym, imitację dotyku przedmiotów czy blokadę położenia dłoni. Te rozszerzenia podstawowego produktu oraz wiele innych zostaną opisane w dalszej części tej pracy. W celu lepszego zrozumienia prezentowanych problemów, zostanie stworzony podstawowy projekt własnej rękawicy-kontrolera, dzięki czemu lepiej będzie można przyjrzeć się budowie takiego urządzenia, elementów wymaganych do jego stworzenia jak i również na podstawie tej pracy możliwość jego odtworzenia i zastosowania w domowych warunkach przy niewielkich kosztach.

## **Rozdział 2**

### **Zagadnienia**

Aby zagłębić się w temat rozszerzonej rzeczywistości trzeba przede wszystkim zrozumieć czym jest rzeczywistość oraz zdefiniować pojęcia które są używane z tym zagadnieniem. Oprócz tego zostanie opisana historia powstania wirtualnych światów, dotycząca zarówno pierwszych urządzeń jak i wyobrażenia wirtualnego świata przez twórców filmów, a także zostanie opisany świat wirtualny z którego każdy może skorzystać i to już od wielu lat. W tym rozdziale opisano szczegółowo rozdzielenie różnych rodzajów rzeczywistości oraz na podstawie zebranych danych zostały wysunięte wnioski dotyczące potencjału technologii.

#### **2.1 Historia kreowania rzeczywistości**

Rzeczywistość według słownika jest to coś co istnieje naprawdę, bądź sytuacja lub warunki, w których ktoś żyje, coś się odbywa [23]. Rozróżnienie tych dwóch definicji jest kluczowe ze względu rozszerzonej rzeczywistości. Rozszerzona rzeczywistość jest to tworzenie nowej formy rzeczywistości poprzez krzyżowanie obiektów rzeczywistych i cyfrowych. Z tego punktu widzenia można zauważyć że według drugiej definicji, rozszerzona rzeczywistość jest częścią rzeczywistości, w szczególności jeśli interakcja pomiędzy światami jest częścią czyjegoś codziennego życia. W przypadku pierwszej definicji, rozróżnienie jest bardziej wy-

---

raźne - to o czym mowa musi być prawdziwe, istnieć naprawdę. Z racji tego że na moment pisania tej pracy, integracja pomiędzy światami wirtualnymi oraz rzeczywistymi nie jest wystarczająco komfortowa aby mówić o użytkowaniu jej na równi z światem rzeczywistym, w dalszej części pracy jako rzeczywistość, przyjmuje się pierwszą definicję [25]. Typy rozszerzonej rzeczywistości wraz z ich definicjami są przedstawione w sekcji 2.3, jednak z perspektywy historii, wszystko zaczęło się od jednego typu nazywanego wirtualną rzeczywistością - w skrócie VR (z ang. Virtual Reality). Pierwszym urządzeniem pozwalającym użytkownikowi na zatrzymanie do wirtualnego świata, a konkretne tworzenie złudzenia przebywania w innym miejscu jest Sensorama. Produkt ten został stworzony w 1962 roku przez Mortona Heiliga. Projekt ten powstał przed grafiką komputerową w związku z czym bazował na wyświetlaniu filmów jako obrazu. Zaledwie trzy lata później Ivan Sutherland nazywany również ojcem grafiki komputerowej, pokazał maszynę do generowania wirtualnej rzeczywistości o nazwie Ultimate Display. Niestety była ona dużych rozmiarów jak i wagi, w związku z czym musiała być przyczepiona do sufitu. W 1977 roku postęp technologiczny trwa, rozwijając nie tylko komputery ale także technologię VR - Dan Sandin stworzył pierwszy kontroler pozwalający użytkownikowi na interakcję ze światem wirtualnym w postaci rękawicy. Jest to początek dla projektów prezentowanych w dalszej części tej pracy. Wraz z latami osiemdziesiątymi powstały pierwsze urządzenia typu Kinect, czyli pozwalające na kontrolowanie oraz interakcję ze światem przy pomocy kamery śledzącej nasze ciało. Jednak dopiero lata dziewięćdziesiąte pozwoliły na użytkowanie technologii w stopniu wystarczającym aby nadawała się on do użytku w branży rozrywkowej. Wirtualna rzeczywistość przeżywa swój prawdziwy rozkwit po raz pierwszy, a to za sprawą powstania firm produkujących pierwsze gogle i rękawice do wykorzystania w VR, zaprezentowaniu pierwszego urządzenia dla konsumentów którzy mogli korzystać ze swojego ciała w świecie

---

wirtualnym przy użyciu wielu czujników oraz zakupie limitowanych zestawów, a także odkrycie potencjału przez branże rozrywkową, próbującą wprowadzić na rynek liczne produkty. Wzmożone zainteresowanie przyczyniło się również do powstania wielu filmów związanych z tym tematem co jeszcze bardziej pogłębiło zainteresowanie wśród ludzi. Wyobrażenia z filmów jednak odstają znaczaco od możliwości technologicznych, a sam rozwój technologii nie jest tak szybki. Z tego oraz wielu innych powodów zainteresowanie normuje się aż do czasu pojawienia się na rynku Oculus Rift w 2011 roku, który na nowo intryguje ludzi oraz pogłębia ogólne zainteresowanie technologią VR. Więcej na temat googli Oculus oraz innych urządzeń aktualnie znajdujących się na rynku jest powiedziane w sekcji 3.1 [32].

## 2.2 Wpływ kultury na rozwój technologii

Od dawna wiadomo że rozwój technologii oraz kultura idą ze sobą w parze. Technologia ma wpływ na twórców oraz artystów, pozwalając im tworzyć nowe, bardziej kreatywne wizje przyszłości, jednocześnie dostarczając lepszych do tego środków, a na podstawie tej twórczości wielu naukowców bazowało się tworząc przełomowe wynalazki, takie jak samo prowadzące się samochody czy telekomunikację cyfrową. Również z technologią VR nie było inaczej. W poniższej sekcji zostanie przedstawione jak twórcy kinematografii prezentowali świat wirtualny, jakie to ma skutki na technologię z której teraz korzystają ludzie a także zostanie pokazany świat wirtualny który był dostępny zanim ludzkość odkryła elektryczność [13].

### 2.2.1 Kinematografia

Wśród pozycji filmowych które zdecydowanie miały wpływ na postrzeganie wirtualnej rzeczywistości znajdują się kilka klasyków. Przede wszystkim warto wspo-

---

mnieć o filmie *Tron*, który zadebiutował w 1982 roku. Film ten pomimo fabułyściśle powiązanej z technologią, pokazujący losy programisty przeniesionego do pamięci komputera. Kolejną ważną rolą tego filmu było szerokie zastosowanie grafiki komputerowej. W latach dziewięćdziesiątych wraz z rozpowszechnieniem się technologii, temat VR stał się dużo bardziej popularny co również można zaobserwować na podstawie tworzonych filmów. To właśnie w tym okresie powstały produkcje takie jak *Johnny Mnemonic*, pokazujący możliwość przenoszenia danych przy pomocy umysłu, *Kosiarz umysłów* - czyli ekranizacja powieści Stephena Kinga, w której wygenerowano komputerowo cyberprzestrzeń, pobudzając wyobrażenie zastosowania tej technologii w rzeczywistości i przede wszystkim *Matrix*. Kultowa pozycja pokazująca ludzi zamkniętej w wirtualnej rzeczywistości którzy sami nie są tego świadomi z powodu realizmu który jest prezentowany - czyli dokładnie to co jest założeniem wirtualnej rzeczywistości. W filmach tych prezentuje się wiele metod łączenia umysłów wraz z technologią co nie wątpliwie było inspiracją dla wielu osób [33]. Z nowszych pozycji niewątpliwie należy wspomnieć o filmie *Ready Player One*, który pokazuje wizję wirtualnego świata, realnego lecz jednocześnie z elementami dostępnymi tylko w środowisku wirtualnym. Film ten bazuje na wielu elementach technologicznych dostępnych w chwili obecnej na rynku, jednak nie zintegrowanych ze sobą a przede wszystkim bez powszechnego dostępu co mogłoby pozwolić na zintegrowanie użytkowników ze światem wirtualnym. Jest on dla wielu wizją tego w jakim kierunku zmierza technologia oraz integracja wielu urządzeń takich jak bieżnie pozwalające na przemieszczanie się w każdym kierunku, realna odczucia całego ciała przy wykorzystaniu odpowiednich kombinezonów no i oczywiście kontrola oraz interakcja przy wykorzystaniu rękawic oraz ciała użytkownika [10]. Wszystkie te wizje sprawiają że użytkownicy pragną coraz bardziej realnych doznań oraz interakcji, a także możliwości osiągnięcia wykonania zadania bez konieczności wychodzenia

---

z domu. Oczywiście realizując to, każdy wie że znajduje się w świecie wirtualnym z powodu świadomego przejścia. Gdyby jednak nawet ktoś się obudził w takiej przestrzeni nieświadomy tranzycji pomiędzy rzeczywistościami, łatwo można to określić na podstawie zamontowanych kontrolerów, kombinezonów czy okularów które wyczuwa się na ciele, a także wad elementów graficznych którym wciąż brakuje wystarczającego realizmu. Istnieje jednak metoda która pozwala temu zapobiec, która może posłużyć jako przykład wirtualnego świata w trybie pojedynczego gracza.

### **2.2.2 Marzenia senne**

Podczas analizy sztucznej rzeczywistości ważnym punktem jest sen, a konkretnie marzenia senne. Przeciętnie człowiek potrzebuje od siedmiu do dziewięciu godzin snu dziennie w cyklu monofazowym, czyli gdy zasypia się i budzi tylko raz dziennie. W tym czasie występują marzenia senna, potocznie nazywane snami. W zależności od osoby swoje sny można pamiętać bądź nie, jednak warto podkreślić że każda osoba ma sny - zapamiętywanie marzeń sennych, tak jak z każdą inną umiejętnością można wytrenować, aby pamiętać więcej szczegółów, miejsc oraz wydarzeń. Mając to na uwadze należy sprecyzować czym one właściwie są. Marzenia senne są serią myśli, obrazów oraz odczuć które dana osoba przeżywa w swoim umyśle w trakcie snu. Nie jest to też dowolny moment w trakcie spania. Sen odbywa się w cyklach, które średnio trwają dziewięćdziesiąt minut i składają się z kilku faz takich jak sen przejściowy, głęboki sen czy faza ruchu gałek ocznych - w każdej fazie można śnić jednak to w tej ostatniej najczęściej występują marzenia senne które są rzeczywiste i realistyczne - w wielu przypadkach nie sposób ich odróżnić od rzeczywistości. Faza ta nazywana fazą REM ( z ang. Rapid Eye Movement) jest etapem snu w którym nasz umysł wprowadzany jest w specjalny stan aby móc osiągnąć ten efekt. Przede wszystkim warto zauważyć

---

że etap ten charakteryzuje się wysoką aktywnością mózgu, porównywalną z tą gdy osoba jest przytomna, sprawia to że bardzo łatwo przerwać ten etap i wybudzić się w trakcie snu. Oprócz tego w naszym mózgu przepływa wiele impulsów przez różne jego obszary, niejako testując połączenia, co jest podejrzewane jako przyczynę tworzenia się w naszym umyśle obrazów, doznań dźwiękowych oraz ruchowych. Aby jednak impulsy nie zostały wysłane domięśni w ciele, w trakcie snu występują ich atonia, czyli zniesienia napięcia mięśniowego. W ten sposób w trakcie gdy ciało leży nieruchomo, w odpowiednim momencie snu dochodzi do symulacji świata w umyśle, w trakcie której osoba może przeżywać wydarzenia które mogą być nawet nie możliwe do zrealizowania w świecie rzeczywistym. Interesującym jest więc fakt, że większość osób zdaje sobie sprawę tuż po przebudzeniu, że wydarzenia które przed chwilą miały miejsce były jedynie marzeniami sennymi - nie były one rzeczywistością. Dzieje się tak za sprawą kory przedczolowej - części mózgu która jest odpowiedzialna za myślenie logiczne, planowanie ruchów i działań, oraz pełni funkcje w działaniu pamięci roboczej. Kora przedczolowa jest jednym z obszarów mózgu który w trakcie snu wytwarza jedynie minimalną aktywność co sprawia że gdy marzenia senne zaczynają się w zupełnie innym miejscu od lokalizacji danej osoby, z ludźmi których dana osoba nie zna, bądź wręcz nie powinno ich tam być, jak na przykład osoby zmarłe bądź znajdujące się w innym miejscu na ziemi, mózg tego nie kwestionuje - przyjmuje że to co się dzieje jest zupełnie normalne. Za sprawą tych mechanizmów, niejako każda osoba codziennie ma dostęp do sztucznej rzeczywistości w której może znaleźć się w dowolnym miejscu, z dowolnymi osobami, przeżywając zdarzenia które mogą być nawet sprzeczne z prawami fizyki.

W obecnych czasach jeżeli mówimy o wirtualnej rzeczywistości, nie myślimy o maszynie takiej jak Sensorama, która jedynie wyświetlała filmy. Poruszając ten temat myślimy o kontrolowanym środowisku w którym użytkownik ma moż-

---

liwość interakcji z otoczeniem a nawet jego kontrole. W rozumieniu marzeń sennych, zwykłe sny są tym czym Sensoroma jest dla współczesnej technologii VR. Świadome śnienie natomiast odkrywa pełną moc możliwości która kryje się w tym naturalnym ludzkim procesie. Aby podjąć próbę świadomego snu należy oczywiście najpierw być w stanie pamiętać swoje zwyczajne marzenia senne. Świadomy sen jest to rodzaj snu w którym osoba śniąca zdaje sobie sprawę z tego że znajduje się w świecie marzenia sennego. Istnieje wiele metod które pozwalają osiągnąć ten stan, jednak ogólnie mówiąc sprowadzają się one do kwestionowania rzeczywistości, dzięki czemu możemy niejako zakodować ten proces w podświadomości. Tak jak wspomniano kora przedczoloowa odpowiadająca za logiczne myślenie nie wykazuje dużej aktywności w trakcie snu, jednak nie jest ona całkowicie nieaktywna, dzięki czemu nadal można ją wykorzystać nawet podczas snu. Jeżeli osoba zakwestionuje rzeczywistość w trakcie marzenia sennego, na przykład poprzez złamanie praw fizyki, i rezultat takiego testu pokaże niemożliwe rezultaty w świecie rzeczywistym, osoba taka odzyskuje świadomość. Często w początkowych próbach dochodzi w tym momencie do wybudzenia, ponieważ jest to niejako przełamanie naturalnego mechanizmu organizmu. Z praktyką jednak dochodzi do stabilizacji świata a dzięki uzyskaniu świadomości, dochodzi do uzyskania kontroli nad naszym umysłem, czyli marzeniem sennym. W ten oto sposób mózg od wewnętrz, bez dodatkowych urządzeń i kontrolerów, generuje własny świat "wirtualny", odtwarzając bodźce wzrokowe, słuchowe, dotyku oraz wszystkie inne zmysły. Osoba kontrolująca ma pełną kontrolę nad własnym ciałem i z jej perspektywy świat w którym się znajduje dostarcza tych samych bodźców co świat rzeczywisty, a dla osób które potrafią kontrolować otoczenie, bodźców możliwych do doświadczenia jedynie poza światem rzeczywistym takich jak np. latanie czy teleportacja [11] [29].

---

## **2.3 Rodzaje rzeczywistości**

Umiejętność świadomego snu jest treningiem przejęcia kontroli nad naturalnym procesem śnienia występującym u każdej osoby, pozwalając niejako doświadczyć tego dokąd technologia wirtualnej rzeczywistości zmierza - bezbłędnego odzwierciedlania świata rzeczywistego, przy dostarczeniu do mózgu wszystkich bodźców w taki sposób aby osoba użytkująca odniosła pełne wrażenie realizmu otoczenia. Mając to na uwadze, w tej sekcji zostanie zdefiniowane czym jest rozszerzona rzeczywistość, jakie rodzaje rzeczywistości obejmuje ten termin oraz wyjaśnić dokładnie pojęcie wirtualnej rzeczywistości które było używane w tym rozdziale.

### **2.3.1 XR**

Akronim XR (z ang. Extended Reality) jest tłumaczony na język polski na wiele sposobów takich jak rzeczywistość rozszerzona ( w odróżnieniu od rozszerzonej rzeczywistości) czy rzeczywistość skrzyżowana. W praktyce jednak najczęściej używa się skrótów angielskich które pozwalają jednoznacznie określić zagadnienie. Termin XR jest używany jako ogólne wyrażenie wszystkich technologii wpływających na zmianę rzeczywistości, które mogą zostać użyte razem bądź pojedynczo. W związku z tym aplikacje mieszające obiekty wirtualne z rzeczywistymi które mogą dowolnie przełączać się pomiędzy tym światem a światem czysto wirtualnym są określane tym terminem, jak i każdy z projektów wykorzystujących tylko jeden typ technologii zmiany rzeczywistości. W chwili pisania tej pracy terminem XR określa się zbiór technologii wirtualnej rzeczywistości, rozszerzonej rzeczywistości a także mieszanej rzeczywistości, jednak określenie to powstało w celu włączenia do jednego określenia również wszystkich terminów które mogą zostać stworzone w przyszłości [20].

---

### **2.3.2 VR**

Pojęcie wirtualnej rzeczywistości określa się skrótem VR ( z ang. Virtual Reality) i definiuje ono część technologii kreowania rzeczywistości w pełni wirtualnej - oznacza to że nie znajdują się w niej żadne elementy świata rzeczywistego. Jeżeli elementy te miały by być prezentowane w takim środowisku muszą one zostać wygenerowane komputerowo. Celem VR jest stworzenie jak najbardziej realistycznych i rzeczywistych odczuć dla użytkownika niezależnie od środowiska w jakim się znajduje. Aby to osiągnąć niezbędna jest symulacja która wpływa na wszystkie zmysły użytkownika. Dlatego też firmy pracują nad coraz bardziej realistycznymi i dokładnymi rozwiązaniami które pozwalają na głębszą imersję w świat wirtualny. Jest to również jednym z głównych powodów dla których zostały stworzone rękawice-kontrolery. Pozwalają one na większą swobodę oraz bardziej naturalny ruch w wygenerowanej przestrzeni, zapewniając lepsze odczucia oraz komfort podczas przebywania w świecie wirtualnym. Nad zapewnieniem realizmu, oprócz metody kontroli środowiska, składa się wiele innych czynników. W tym celu powstają specjalne kombinезony które odwzorowują ruch całego ciała, bieżnie które pozwalają się poruszać w VR, dzięki naturalnemu przemieszczeniu się, które za sprawą bieżni odbywa się w miejscu w świecie rzeczywistym, okularów które pozwalają wyświetlać przed oczami środowisko wirtualne, starając się odwzorować głębię oraz wiele innych urządzeń. Ilość urządzeń oraz metod imitacji pokazuje jak skomplikowanym problemem jest odwzorowanie rzeczywistości, a wszystko to musi odbywać się niezwykle płynnie ponieważ mózg człowieka nie rozróżnia rzeczywistości od świata wirtualnego. Oznacza to że jeżeli bodźce wzrokowe nie pokrywają się z bodźcami słuchowymi dochodzi do tak zwanego zjawiska choroby lokomocyjnej. Sprawia to że przebywania w wirtualnym świecie przez długi okres czasu może doprowadzić do dyskomfortu dla użytkownika.

---

Aby zapobiec tego rodzaju problemom równie ważnym jest dokładna znajomość fizjologii człowieka, co prowadzi do współpracy wielu działów nauki w celu stworzenia prawdziwie realistycznych produktów. Pomimo wielu przeszkód jak przedstawiono w sekcji 2.1, wirtualna rzeczywistość była pierwszą wizją naukowców którą starano się zrealizować - pomimo dużych nakładów pracy w celu zapewnienia realistyczności, niesie ona za sobą wiele możliwości zarówno w biznesie jak i życiu codziennym, co sprawia że dla wielu firm jest to możliwość realizowania własnej wizji wirtualnej przyszłości [20] [30].

### 2.3.3 AR

Kolejnym terminem wchodzącym w skład XR jest rozszerzona rzeczywistość, bądź też nazywana rzeczywistością poszerzoną. Skrótnie zapisuje się ją jako AR (z ang. Augmented Reality). Podobnie brzmiąca do XR w szczególności w języku polskim jest częstym powodem dla którego powszechnie używa się skrótów angielskich mówiąc o tym obszarze technologii. AR jak nazwa wskazuje poszerza rzeczywistość zamiast ją zastępować tak jak to miało w przypadku VR. Oznacza to że do elementów świata rzeczywistego zazwyczaj pozyskanych przy pomocy obrazu z kamery, wyświetlanych poprzez wybrane źródło obrazu o których mowa w rozdziale 3, są dodawane elementy stworzone przy pomocy grafiki komputerowej, zazwyczaj trójwymiarowe. Technologia ta dzięki wykorzystaniu w dużej mierze świata rzeczywistego jest łatwiejsza w użytkowaniu a jej zastosowanie jest powszechnie stosowane w wielu branżach. W szczególności duży sukces technologia ta odniosła gdy wykorzystano ją do stworzenia popularnej gry na smart-fony *Pokemon Go*, która przy wykorzystaniu geolokalizacji oraz AR zapewniała realistyczne doświadczenia łapania stworzonych trójwymiarowych postaci w świecie rzeczywistym [20]. Podgląd interfejsu aplikacji obrazujący wykorzystanie technologii AR w *Pokemon Go* pokazuje rysunek 2.1.



Rysunek 2.1: Interfejs aplikacji *Pokemon Go* wykorzystujący AR.

Źródło: <https://gameradar.pl/aktualizacja-pokemon-go-utrudnia-lapanie-pokemonow/>

### 2.3.4 MR

Mieszana rzeczywistość, czyli MR (z ang. Mixed Reality) jest terminem który prawdopodobnie jest najczęściej błędnie używanym wśród zagadnień związanych z XR. Mieszana rzeczywistość podobnie jak AR wykorzystuje do działania zarówno świat rzeczywisty jak i stworzony komputerowo, jednak w odróżnieniu od AR nastawiony jest głównie na interakcję pomiędzy tymi światami. Obiekty tworzone cyfrowo wyświetlane w świecie rzeczywistym mogą być modyfikowane i wpływać na środowisko rzeczywiste które jest wyświetlane jak i obiekty rzeczywiste mogą zostać przeniesione do świata wirtualnego. Interakcja ta i zastosowanie jest niejako technologią przyszłości, wizją producentów, ponieważ do tej pory nie ma łatwo dostępnych produktów które by wykorzystywały ten rodzaj XR, natomiast istnieją takie rozwiązania dla biznesu oraz w ośrodkach badawczych, które prawdopodobnie staną się bardziej dostępne wraz z rozwojem tej technologii oraz ulepszonych rozwiązań w innych dziedzinach technologicznych takich jak grafika komputerowa, prędkość przesyłania czy przetwarzania danych [20].

---

## **2.4 Potencjał technologii**

Rozwiązania XR przebyły długą drogę od czasu powstania pierwszej wizji tego typu rozwiązania. Obecnie istnieje wiele urządzeń które powszechnie korzystają z XR zarówno na użytku prywatny, branży rozrywkowej oraz w celu usprawnienia procesów biznesowych. Niemniej jednak dopiero są opracowywane rozwiązania które mogłyby zapewnić większą przenośność tego typu urządzeń, sprawniejsze połączenie i przede wszystkim realizm. Wykorzystanie w życiu codziennym staje się coraz bardziej powszechnne, w związku z tym można też się spodziewać ulepszeń produktów zarówno pod względem sprzętu jak i oprogramowania a także dostępności tych produktów przy mniejszym koszcie. Można być pewnym że technologia ta nie przestanie się rozwijać, znajdując coraz to nowe zastosowania, inwestorów a także twórców którzy wprowadzając większą liczbę produktów na rynek, tworzą większe zainteresowanie wśród społeczności. Pytanie stawiane przez technologię XR to nie czy ta technologia ma przyszłość, a kiedy stanie się ona powszechnie stosowana. W dalszych częściach tej pracy zostaną pokazane sposoby w jakich obecnie przedstawiane i wykorzystywane są opisywana do tej pory technologie, a także zostanie szczegółowo opisany temat rękawic-kontrolerów, urządzenia które nie jest standardem wśród kontrolerów na rynku dla pojedynczych konsumentów VR, jednak już jest wykorzystywany w biznesie a także przewidywany jest jako kontroler przyszłości [20].

## Rozdział 3

# Prezentacja kreowanej rzeczywistości

### 3.1 Okulary VR

### 3.2 Smart-fon jako urządzenie do prezentacji

### 3.3 Kontrolery i IoT

#### 3.3.1 Problemy kontrolerów

#### 3.3.2 Działanie czujników

IMU

### 3.4 Dłoń jako kontroler

historia, pierwsze firmy, popularne rozwiązania

## **Rozdział 4**

# **Rękawice jako kontrolery**

Kontroler w postaci rękawicy jest pojęciem często używanym ze względu na popularność tego rozwiązania. Rękawiczka zakładana na dłoń z zamontowanymi czujnikami dzięki którym deweloper uzyskuje wszelkie potrzebne informacje aby wchodzić w interakcję z otoczeniem. Idea która za tym stoi nie dotyczy jednak rękawiczki samej w sobie - chodzi o wykorzystanie ludzkiej dłoni jako kontrolera a rękawiczka jest używana jako nośnik temu służący. Ze względu na różne podejścia istnieją typy kontrolerów wykorzystujących dłoń do obsługi świata, które zostaną opisane w dalszej części tego rozdziału. Kontrolery a w szczególności dlonie są głównie wykorzystywane w świecie VR, w pozostałych rodzajach XR wystarczyłaby analiza obrazu na podstawie której określa się interakcję użytkownika z otoczeniem. W przypadku VR które przenosi użytkownika do zupełnie innego świata, wykorzystanie kontrolerów wydaje się więc obowiązkowe.

### **4.1 Rozwój rękawicy jako kontrolera**

Pomysł wykorzystania dloni jako kontrolera, pomimo wielu lat od pierwszego wdrożenia wciąż znajduje zastosowania głównie w biznesie, jednak produkty te są dostępne również dla zwykłych konsumentów. Problemem jest jednak cena która czasami wynosi tyle co zestaw do obsługi VR bądź więcej. Oprócz tego

---

nie ma zbyt wielu pozycji w sklepie które by wykorzystywały w sposób szcze-gólny tego rodzaju kontrolery. Dlatego też to właśnie głównie firmy są zaintere-sowane tego typu produktami, które dla specyficznego przypadku tworzą aplika-cje obsługujące tego typu kontrolery. Wykorzystanie XR i rękawic kontrolerów w biznesie pokazano w rozdziale 5. Jednak aby produkt ten był w stanie speł-nić oczekiwania rynku musiał przejść długą drogę od projektu stworzonego przez Dana Sandina. Jego rękawica *The Sayre Glove* była w pełni przewodowa a jedy-nymi czujnikami które obsługiwała był pomiar wygięcia palców, służący głównie do zmiany pozycji suwaków. Była jednak tania i lekka co spełniało podstawowe oczekiwania w tamtych czasach. Rękawice tą przedstawia rysunek 4.1. W 1983



Rysunek 4.1: Pierwsza rękawica VR - Sayra Glove

Źródło: <https://www.evl.uic.edu/entry.php?id=2162>

roku Gary Grimes stworzył kolejny produkt warty uwagi - była to rękawica która potrafiła rozpoznawać osiemdziesiąt specjalnych gestów, które następnie były za-mieniane na litery alfabetu. Produkt ten jednak nie został nigdy dostępny do kupienia. Kilka lat później, w 1987 roku nastąpił przełom w dziedzinie ręka-wic kontrolerów. Została stworzona rękawica która potrafiła monitorować sześć

---

stopni swobody - oznacza to przemieszczenie oraz orientację wokół trójwymiarowego układu współrzędnych. Wykonana została przez Thomasa Zimmermana wraz ze współpracownikami i została spopularyzowana i rozpowszechniona na całym świecie. Niewątpliwie był to największy przeskok w rozwoju tego urządzenia od czasu pierwszego projektu. Dwa lata później powstała rękawica która jako pierwsza była dostępna w celach rozrywkowych. Stworzona przez firmę zabawkową Mattel, dla gier konsolowych produkowanych przez Nintendo, rękawica nazwana *The Power Glove*. Produkcja jednak została przerwana po dwóch latach od wprowadzenia na rynek produktu. Są to wybrane pozycje które w jakimś stopniu zmieniły rynek oraz wprowadziły zmiany w tym segmencie. Wraz z dalszym rozwojem wirtualnej rzeczywistości powstały kolejne produkty a w szczególności po roku 2000, podczas pierwszej fali popularności technologii VR powstało wiele dopracowanych produktów [16].

## 4.2 Podział rękawic-kontrolerów

Rozwój rękawic-kontrolerów wprowadził wiele zmian począwszy od technologii na której bazują, połączeń które używają jak i funkcjonalności które są w stanie zapewnić. Na tej podstawie nadal większość produktów stara się wyróżnić na tle konkurencji jednak bazowe podejście do produktu sprawia że można go podzielić według ogólnych kategorii, do których można zaliczyć rękawice klasyczne, systemy nasadowe oraz egzoszkielety. W dalszych częściach tej sekcji zostaną opisane szczegółowo te typy wraz z prezentacją współczesnych rozwiązań które są dostępne na rynku [17].

### 4.2.1 Klasyczne

Pierwsze, najstarsze typy rękawic określane są jako klasyczne ponieważ przypominają one zwykłe rękawiczki używane w życiu codziennym z tym że podpięto do

---

nich różnego rodzaju sensory, mikrokontrolery, przekaźniki danych i baterie. Innymi słowy sprawiono że zwykła rękawiczka jest w stanie zbierać dane na temat położenia dloni i palców użytkownika, a następnie bezprzewodowo przesyłać te dane do odbiornika, zapewniając przy tym wygodę pracy oraz ponad kilka godzin użytkowania na jednym ładowaniu w zależności od produktu. Prawdopodobnie najbardziej znanym produktem na rynku są rękawice należące do firmy Manus Vr, która dostarcza kilka rozwiązań w zależności od dodatku za które użytkownik chce dopłacić. Wykonane z elastycznego materiału, lekkie i cienkie dzięki czemu są przyjemne w używaniu, z materiałem odciętym w czubkach palców dla większej wygody. Waga to około 70 gramów, co sprawia że użytkowanie to czysta przyjemność. Rękawice łączą się z takimi zestawami VR jak Oculus czy Vive i są szeroko wykorzystywane na rynku biznesowym przez takie firmy jak Audi, BMW, Ubisoft czy NASA. Cena za rękawice to od 3000 do 4000 euro - zdecydowanie nie należą one do najtańszych. Od strony technicznej, w początkowej wersji tego projektu zostały zastosowane czujniki oporu, które zwracały różny opór, jednak zastąpiono je w nowszej wersji zestawem czujników zaprojektowane przez firmę Bosch w skład których wchodzą jedenaście czujników na palcach, dwa na każdym poza kciukiem gdzie znajdują się trzy - żyroskop, akcelerometr i magnetometr dla dokładnego mierzenia położenia naszej dłoni. Czujniki te są elastyczne i działają analogowo sprawdzając knykcie oraz pierwszy staw palca a także od knykcji do drugiego stawu palca łącznie dając dziesięć stref pomiaru. Czujniki orientacyjne znajdują się na wierzchniej części dłoni oraz kciuka - pomiar jest dokonywany dzięki płytce IMU zapewniającej jedenaście stopni swobody. Czujniki na palcach wskazują dokładność  $\pm 3$  stopni. System śledzenia natomiast oblicza pozycję dłoni w przestrzeni dzięki użyciu innych systemów takich jak Xsens, Vive Tracking, OptiTrack, Vicon czy ART. Obsługują one pracę wielu użytkowników jednocześnie w jednym świecie wirtualnym. Bateria została wyprodukowana przez

---

firmę Varta a producent szacuje czas pracy na baterii pomiędzy 4 a 6 godzinami intensywnego użytkowania. Rękawice działają bezprzewodowo przy opóźnieniu poniżej 5 milisekund. Wspierają one również technologię wibracji, dzięki czemu można samodzielnie zaprogramować moduł wibrujący dla odpowiednich czynności świata wirtualnego. Firma wypuściła własne wtyczki dla takich środowisk jak Unity, Unreal Engine oraz Autodesk MotionBuilder. Jest to jedno z najbardziej rozbudowanych rozwiązań na rynku które wspiera rozwój swojego oprogramowania na wielu platformach i stanowi jednego z największych konkurentów dla pozostałych produktów [34]. Rękawice te widoczne są na rysunku 4.2a. Podobnym produktem występującym na rynku są rękawice firmy Noitom o nazwie *Hi5*. Rękawice te kosztują znacznie mniej, ich cena to około 1000\$, jednak posiadają znacznie więcej wad. Przede wszystkim nie wspierają one zestawu Oculus Rift. Tak jak w poprzednim przypadku, bazują one na kontrolerach ruchu dostarczonych przez zewnętrzną firmę - w tym przypadku są to kontrolery Vive. Źródłem zasilania są zwykłe baterie AA co stanowi poważny problem dla dłuższego użytkowania. Rękawice wspierają *Haptic Feedback*, czyli system wibracji oparty na interakcji z przedmiotami w świecie wirtualnym, zapewniają śledzenie wszystkich palców u dłoni oraz używają czujników takich jak żyroskop, akcelerometr i magnetometr wbudowanych w jednostkę IMU. Widać więc że spełniają one stawiane przed tymi kontrolerami oczekiwania jednak posiada wiele niedociągnięć które potencjalnie zniechęcają do kupna tego produktu [22]. Rysunek 4.2b przedstawia omawiany kontroler. Kolejnym produktem jest *Exoglobe*, który wyróżnia się na tle konkurencji dzięki zastosowania technologii ultradźwięków. W ten sposób śledzi dłoń w prawie  $180^\circ$  z dokładnością nawet do 0,5 mm. W oprogramowaniu rękawic znajduje się moduł sztucznej inteligencji który uczy się gestów, sposobu poruszania się oraz obszaru w którym porusza się użytkownik. Wszystko zapewniane jest za cenę 99\$ która jest zdecydowanie mniejsza od propozycji kon-



(a) Rękawica firmy Manus Vr

Źródło: <https://manus-vr.com/mocapgloves>

(b) Rękawica firmy Noitom

Źródło: <https://hi5vrglove.com/store/hi5glove>

Rysunek 4.2: Przykładowe klasyczne rękawice VR.

kurencji [6]. Avatar VR jest to produkt hiszpańskiej firmy NeuroDigital, która stworzyła rękawicę wspierającą haptic feedback, wykonaną z oddychającego materiału który również jest materiałem przewodzącym co pozwoliło zaimplementować sterowanie gestami. Za cenę 1500\$ dostajemy parę rękawiczek wraz z odbiornikiem USB który łączy się poprzez Bluetooth 4.0. Produkt zapewnia od sześciu do ośmiu godzin pracy na baterii po której należy naładować akumulator poprzez 5 V micro USB. Do rękawiczek można podłączyć odpowiednie czujniki które należy podpiąć na ramieniu oraz klatce co zapewnia śledzenie całej górnej części ciała. Sterowanie zapewniane jest poprzez 9 osiowe IMU. Rękawiczki mają wbudowanych 10 płyt wibrujących które zapewniają 1024 różnych poziomów wibracji. W rękawiczkach znajdują się 4 strefy przewodzące (dłoń, kciuk, palec wskazujący oraz środkowy). Produkt posiada również technologię NZDE

---

(z ang. Near zero drift Experience), która pozwala powtarzać te same gesty w celu eksperymentowania w środowisku wirtualnym bez wprowadzania kumulacji znaczących błędów które gromadzą się w szczególności przy korzystaniu z danych żyroskopu [21]. Ostatnim produktem z kategorii klasycznych rękawic który warto wymienić są rękawice CaptoGlove. Firma produkująca ten kontroler rozszerza swoją funkcjonalność poza świat VR. Producent daje możliwość używania swoich rękawic poprzez technologię plug and play (z ang. podłącz i używaj) z komputerami oraz smart-fonami. Dzięki użyciu zapewnianego sdk można dalej przenieść nasze doświadczenia na platformy VR, konsole do gier czy kontrolować przy ich użyciu drony, roboty czy telewizory. Za pojedynczą rękawicę klient zapłaci 250\$ za parę natomiast 490\$. Aby używać produktu w VR, grach video czy na smart urządzeniach należy dokupić Capto sensor czyli czujnik ruchu pozwalający określić położenie dłoni w przestrzeni kosztujący dodatkowe 190\$. Urządzenia kontrolujemy przy użyciu gestów dłoni, odpowiednie ruchy są tłumaczone na akcji wykonywane na różnych platformach. Na czubkach palców został umieszczony materiał pozwalający kontrolować ekrany dotykowe bez zdejmowania kontrolera. Rękawica łączy się z wymienionymi urządzeniami przy pomocy Bluetooth i według producenta pozwala na pracę przez około 10 godzin. Jest to kolejna pozycja wśród rękawic ciesząca się popularnością i jest wykorzystywana przez takie firmy i organizacje jak Mitsubishi, U.S. Air Force czy Fujitsu [8].

#### **4.2.2 Nasadowe**

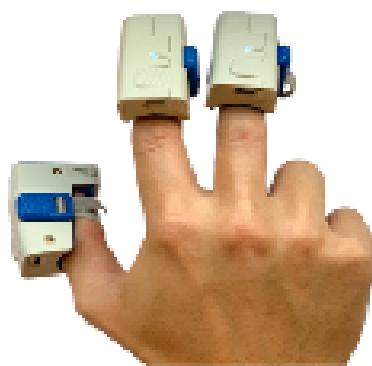
Kolejnym typem urządzeń pozwalających na używanie dłoni jako kontrolera są urządzenia nasadowe. Nie są to typowe rękawiczki, a jedynie jej elementy których można by się spodziewać na czubkach palców, nadgarstkach czy też dloni, które połączone odpowiednio ze sobą pozwalają na odczytywanie danych dotyczących położenia i orientacji dłoni, palców, zapewniają wibracją oraz innego ro-

---

dzaju imitacje świata rzeczywistego. W tym segmencie zdecydowanie wyróżniają się nakładki na palce, które skupiają się na symulacji dentyku, a w szczególności dwie firmy które się tym zajmują - Tactai z nakładką na palec nazwaną Tactai Haptic Module oraz GoTouchVR ze swoim produktem VrTouch. Pierwsza z tych firm zajmuje się stworzeniem małego urządzenia zakładanego na czubki palców które pozwala poczuć świat wirtualny. Firma ta zajmuje się badaniem tego jak ludzie odczuwają różne przedmioty w świecie rzeczywistym poprzez zmysł dentyku i stara się imitować te same odczucia dla zdarzeń zachodzących w świecie wirtualnym. Poprzez specjalne urządzenie przypominające długopis, rejestruje wibracje różnych materiałów i ich tekstury, następnie tworzy w systemie wzory wibracji, dokładając do tego dźwięki, jakie wspomniane materiały wydają podczas poruszania po nich urządzeniem badawczym. Następnie na podstawie zebranych danych, imitująto samo zjawisko gdy użytkownik wchodzi w interakcje z danym materiałem w świecie wirtualnym. Oprócz tego znajduje się tam również mechanizm pozwalający na śledzenie palca. Konstrukcja ta pokazana jest na rysunku 4.3a [31]. Druga firma obrała inny kierunek przy tworzeniu produkty, ich nakładki na palce są rozszerzeniem które można wykorzystać z już istniejącymi produktami rękawiczek, dzięki czemu można wykorzystać wbudowane w nie systemy śledzenia. Firma ta współpracuje z urządzeniami firm takich jak Manus czy Noitom. Ich urządzenia współpracuje z trzema nakładkami jednocześnie dla każdej z dloni przy zapewnieniu niskiego poziomu opóźnienia. Poza tym ich funkcjonalność jest taka sama jak prezentowanej w tej sekcji konkurencji - pod względem wyglądu natomiast urządzenia ta przypominają nie wyglądają zbyt nowocześnie. GoTouchVr współpracuje głównie w ramach sektora biznesowego i medycznego dostarczając swoich rozwiązań do takich firm jak BMW czy Medtronic. Urządzenie to pokazane jest na rysunku 4.3b [15].



(a) Urządzenie Tactai  
Źródło: <https://www.tactai.com/company>



(b) Urządzenie TouchVr  
Źródło: <https://www.gotoucdrv.com/copy-of-technology-devices-1>

Rysunek 4.3: Urządzenia nasadowe.

#### 4.2.3 Egzoszkielety

dextarobotics - dexmo pierwsze exo vrglув Cyber Grasp Haptx Cynteract

Powstawały kolejne projekty takie jak chińska firma Dexta Robotics która wystartowała z kampanią finansowania społecznego w 2014 roku, jednak zbiórka pieniędzy zakończyła się niepowodzeniem. Była to jedna z pierwszych spopularyzowanych egzoszkieletów, która pomimo porażki w zbiórce pieniędzy przetrwała i firma dokończyła projekt na własną rękę w 2016 roku.

## **Rozdział 5**

### **Zastosowania w biznesie**

**5.1 Lotnictwo i symulatory**

**5.2 Medycyna**

**5.3 Architektura**

**5.4 Rozrywka**

# Rozdział 6

## Projekt rękawicy

Istotą poniższego rozdziału jest pokazanie użytych w projekcie podzespołów i technologii oraz lepsze zrozumienie powodów dla których to właśnie te produkty zostały wybrane. Po zapoznaniu się z motywem, zostanie szczegółowo opisana specyfikacja tych produktów a także sposób ich poskładania w spójną całość, co sprawiło pewne problemy względem oryginalnego szkicu - próby oraz efekty rozwiązywania tych problemów również zostaną opisane w tym rozdziale. Po nakreśleniu podstawowych założeń projektu zostanie zaprezentowana finałowa wersja, a także szczegółowo zostanie omówiony kod rękawicy-kontrolera, który jest obsługiwany przez mikrokontroler i stanowi najważniejszą część tego projektu.

proof read

Opis: BLE, Rotacja, przesunięcie i animacja, przy opisywanych sensorach

### 6.1 Przegląd podzespołów użytych w projekcie

Jak dowiedzieliśmy się z rozdziału ?? dotyczącego podstawowych komponentów rękawic, kluczowym dla powodzenia projektu jest ustalenie następujących pozycji:

- orientacji dłoni względem punktu początkowego
- położenia względem punktu początkowego
- moment i stopień zgięcia palców

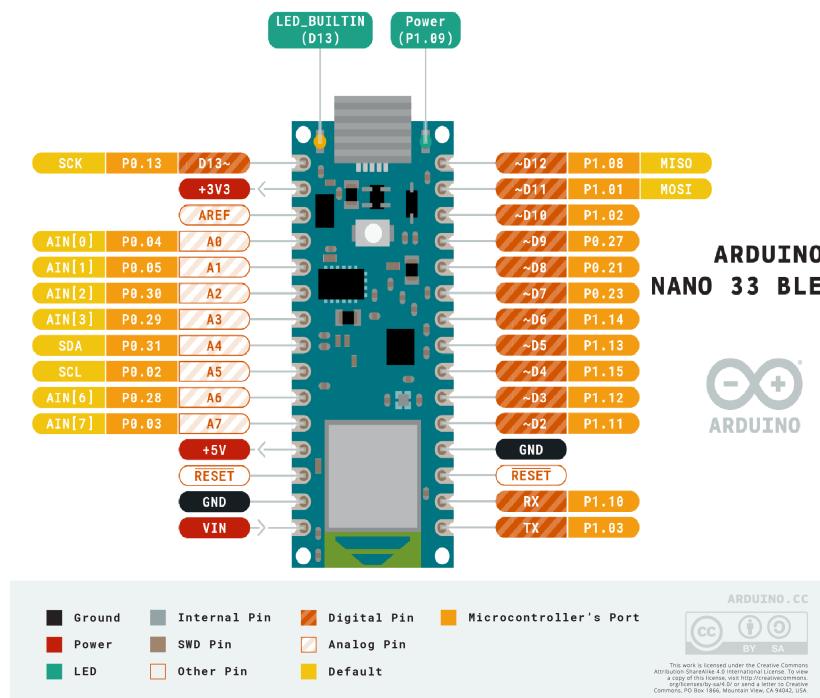
---

W tym celu należy zebrać informacje z czujników, a następnie wszystkie te informacje należy przesyłać do pożądanego urządzenia. Elementem które pozwala to osiągnąć w tym projekcie jest mikrokontroler Arduino nano 33 BLE, który odpowiada za dostarczenie informacji z żyroskopu, akcelerometru a także czujników wygięcia. Zasady działania pierwszych dwóch zostały opisane z podrozdziale ???. Natomiast w poniższych podrozdziałach zostanie opisane rozwiązanie zastosowane do odczytu położenia palców, zasada działania przy wykorzystaniu rezystorów oraz sposób połączenia wszystkich wspomnianych elementów w jeden finałowy kontroler.

### **6.1.1 Mikrokontroler**

Jak przed chwilą wspomniano, w projekcie wykorzystywana jest płytka od Arduino, która nosi nazwę Nano 33 BLE. Jest to małych rozmiarów płytka o wymiarach 45 x 18 mm, pozwalająca na wysoką wydajność przy jednoczesnym małym poborze prądu, co zapewnia użyty mikrokontroler nRF52480 o taktowaniu 64 MHz. Do dyspozycji mamy również pamięć RAM o pojemności 256 kB oraz pamięć Flash o pojemności 1 MB. IMU które zostało zamontowane na płytce to LSM9DS1, które obsługuje akcelerometr, żyroskop oraz magnetometr w trzech osiach. Więcej informacji na temat IMU jest przedstawione w części 6.3. Warto na wstępie zauważyć że Nano 33 BLE pracuje domyślnie wyłącznie z napięciem 3,3 V, w związku z czym nie należy podłączać bezpośrednio zasilania o większym napięciu. W celu podłączenia zasilania 5 V należy zlutować zworkę znajdującą się pomiędzy pinami RDT oraz A7 - temat ten nie zostaje jednak poruszony w tej pracy, ponieważ na potrzeby projektu używane jest zasilanie poprzez złącze micro USB które to również jest obsługiwane. Płytki ta posiada wiele użytecznych sensorów, jednak na potrzeby tej pracy została wybrana z powodu wbudowanej inercyjnej jednostki pomiarowej, dzięki czemu można było uprościć konstrukcję

oraz zmniejszyć ilość połączeń na rękawicy, wbudowanego modułu Bluetooth - a w tym przypadku modułu Bluetooth Low Energy obsługiwany w standardzie 5.0, a to wszystko w przystępnej cenie co również było jednym z kryteriów przy tworzeniu tego projektu. Płytkę w momencie tworzenia tej pracy można kupić za 119 zł. Wartym uwagi jest fakt możliwości zakupu płytki bez wyprowadzonych złącz, co w przypadku opisywanego projektu pozwoli na zmniejszenie wymiarów oraz większą swobodę montażu [4].



Rysunek 6.1: Opis wejść oraz wyjść Arduino Nano 33 BLE

Źródło: [https://content.arduino.cc/assets/Pinout-NANOble\\_latest.pdf](https://content.arduino.cc/assets/Pinout-NANOble_latest.pdf)

Z najważniejszych elementów układu płytka należy wiedzieć że posiada ona dwie diody po dwóch stronach portu micro USB - zielona indykuje podłączone zasilanie, natomiast pomarańczowa zaczyna mrugać gdy jest przesyłany kod do mikrokontrolera. Oprócz tego do dyspozycji są dwa piny wyjściowe zasilające o

---

napięciu 3,3 V oraz 5 V, jeden pin zasilający wejściowy, którego ograniczenia zostały wspomniane w poprzednim paragrafie, a także dwa piny uziemiające, po jednym z każdej strony płytki. Posiada ona piny zarówno analogowe jak i cyfrowe, jednak na potrzeby tego projektu zostały użyte jedynie piny analogowe, których do dyspozycji jest aż osiem umiejscowionych po jednej stronie, przy czym warto zwrócić uwagę że piny A4 oraz A5 używane są jako magistrala I2C w związkach z czym zalecane jest nie stosowanie tych wejść analogowych. Szczegółowy opis wejść/wyjść płytki przedstawiony jest na grafice 6.1. W tym projekcie wykorzystano zasilanie poprzez złącze micro USB, wyjście o napięciu 3,3 V w celu uzyskania odczytów z czujników wygięcia na pinach analogowych A0, A1, A3, A6 oraz A7, o których zostanie więcej powiedziane w punkcie 6.2, a także uziemienie znajdujące się po tej samej stronie płytki.

### **6.1.2 Czujnik wygięcia**

Kluczowym dla działania kontrolera jest możliwość określenia pozycji palców względem dłoni. Ma to wiele zastosowań zarówno wizualnych jak i praktycznych. Ważne jest aby odwzorować świat rzeczywisty tak dokładnie jak to możliwe - im lepsze odwzorowanie tym bardziej zmysły użytkownika zostaną oszukane, podwyższając komfort użytkowania technologii wirtualnej rzeczywistości. Za stroną praktyczną natomiast przemawia możliwość śledzenia palców w celu dokładnego ich użycia w stworzonym środowisku np. do ściskania i podnoszenia obiektów czy też korzystania z klawiatury wirtualnej. Jak wspomniano w rozdziale ?? dotyczącym komponentów komercyjnych rękawic - wśród znanych producentów na rynku, decyduje się na użycie wielu inercyjnych jednostek pomiarowych, na podstawie których są w stanie dokładnie określić położenia każdej części palca, bądź też nie aż tak popularne rozwiązanie, które wykorzystujące specjalistyczne sensory służące do pomiaru stopnia wygięcia czujnika względem

---

pozycji prostej. Czujnik ten po podłączeniu do prądu zwiększa swój opór wraz ze zwiększym stopniem odchylenia. Oba te rozwiązania pomimo wysokiej dokładności pomiarów nie są rozwiązaniami tanimi. W związku z tym na potrzeby stworzenia taniego kontrolera, należało znaleźć rozwiązanie bardziej przystępna a jednocześnie pozwalające na osiągnięcie tego samego celu.

Aby osiągnąć postawione założenia zostały skonstruowane czujniki wygięcia w domowych warunkach. Rozwiązanie to jest często używane do osiągnięcia pomiaru stopnia wygięcia bez konieczności wydawania ponad 100 zł na jeden czujnik [5]. Jest ono stosunkowo proste w założeniach i wymaga jedynie dwóch klużowych elementów. Tkaniny przewodzącej o specjalnych właściwościach oraz dwóch przewodników po obu stronach materiału. Z jednej strony zostanie podłączone napięcie z drugiej natomiast uziemienie. Ważnym jest aby połączenia te się ze sobą nie stykały w żadnym punkcie a jedynie zostały nałożone na siebie, z tkaniną ściśniętą pomiędzy nimi - dzięki temu mamy pewność że odczyty które otrzymamy będą prawidłowe. Pozostaje odpowiedzieć na pytanie jakiego rodzaju materiał należy wykorzystać. Na rynku znajdziemy wiele rodzajów materiałów które zmieniają swój opór w zależności od spełnienia takich kryteriów jak nacisk, temperatura, rozciągnięcie czy też właśnie zgięcie materiału. Pomimo próby uzyskania materiału który zmienia swój opór w zależności od rozciągnięcia, co pozwoliłoby na skonstruowanie części na palce rękawiczki z tego materiału, zapewniając dokładniejszy i bardziej estetyczny efekt końcowy, w momencie projektowania rękawicy był on jedynie możliwy do sprowadzenia ze stanów, co nie było najtańszym rozwiązaniem. W związku z tym zdecydowano się na użycie foli Velostat która jest czuła na nacisk or zginanie [2]. W roli przewodnika wybrano nić przewodzącą, która zapewniła potrzebną elastyczność oraz możliwość przy mocowania poszczególnych elementów przy jednoczesnym zapewnieniu funkcjonalności. Elementy te zostały sklejone na kawałku taśmy samoprzylepnej oraz do-

---

datkowo sklejone przy brzegu aby nie wyśliznęła się w trakcie korzystania z czujnika. Efekt końcowy jest widoczny na zdjęciu 6.2. W celu otrzymania pomiarów wszystkich palców zostało wykonanych pięć takich sensorów, o szerokości 15 mm; dwa o długości 8 cm, dwa o długości 10 cm a także jeden 11 cm, w celu jak najlepszego dopasowania względem miejsca na palce na zakupionej rękawicy do której sensory zostaną przymocowane, co można zobaczyć na zdjęciu 6.6.



Rysunek 6.2: Sensor wygięcia własnego wykonania

### 6.1.3 Rezystor

Rezystor, potocznie zwany opornikiem, jest to prosty element elektroniczny, posiadający jedynie wyjścia z dwóch stron elementu łączącego. Element ten tworzy opór, powodując ograniczenie przepływającego przez niego prądu gdy jest włączony do obwodu szeregowo. Opór ten jest mierzony w omach. Istotną informacją jest fakt że nadmiar prądu jest zamieniany przez opornik na energię cieplną, a także brak zdefiniowanego kierunku - co oznacza że działa on niezależnie od sposoby zanotowania go w układzie.

Pomimo swojej prostoty budowy i zastosowania, dla danego projektu ważne

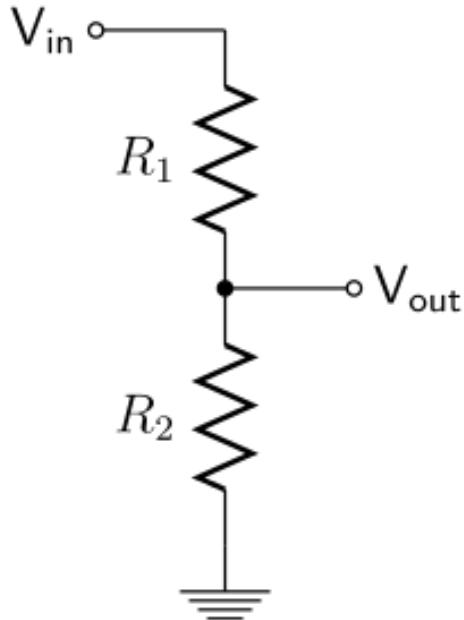
---

jest aby wybrać odpowiednie rezystory. Podstawową wartością na którą należy zwrócić uwagę jest rezystancja. Rezystancję podaje się w omach i można spotkać na rynku zakres od milionów do megaomów. Spośród dostępnych rodzajów rezystorów w projekcie zostały użyte rezystory THT (z ang. Through-Hole Technology) - czyli tak zwane rezystory do montażu przewlekłanego. W tym rodzaju oporników rezystancja jest ilustrowana poprzez kolorowe paski umieszczone wokół oporu, co pozwala odczytać ich wartość według ilustracji 6.3. Alternatywą do tego sposobu jest podłączenie rezystora pod miernik elektryczny ustawiony w tryb pomiaru oporu [12]. Element ten jest kluczowy w celu ograniczenia przepływu prądu w obwodzie rękawicy, co pozwala na monitorowanie oporu wytwarzanego poprzez czujnik wygięcia.

Kolor	Wartość		Mnożnik	Tolerancja ± %	Współczynnik temp. ± ppm/K
	1 pasek	2 pasek			
czarny		0	x 1 Ω		250
brązowy	1	1	x 10 Ω	1	100
czerwony	2	2	x 100 Ω	2	50
pomarańczowy	3	3	x 1 kΩ		15
żółty	4	4	x 10 kΩ		25
zielony	5	5	x 100 kΩ	0,5	20
niebieski	6	6	x 1 MΩ	0,25	10
fioletowy	7	7	x 10 MΩ	0,1	5
szary	8	8	x 100 MΩ	0,05	1
biały	9	9	x 1 GΩ		
złoty			0,1 Ω	5	
srebrny			0,01 Ω	10	
brak				20	

Rysunek 6.3: Oznaczenia rezystorów

Źródło: <https://sites.google.com/site/informatykauni/jna/home/poszczegolne-czesci/rezystor>



Rysunek 6.4: Układ dzielnika napięcia  
Źródło: [https://www.wikiwand.com/en/Voltage\\_divider](https://www.wikiwand.com/en/Voltage_divider)

Sposób działania układu, nazywanego dzielnicą napięcia jest pokazany na rysunku 6.4, oraz wyraża się wzorem

$$\nu_{out} = \nu_{in} \left[ \frac{R_2}{R_1 + R_2} \right]$$

Wzór ten podaje napięcie wyjściowe  $\nu_{out}$ , które równa się napięciu wejściowemu  $\nu_{in}$  przeskalowanemu przez stosunek rezystorów. W opisywanym przypadku jest to stosunek zastosowanego rezystora  $4.7k\Omega$  wyrazonego we wzorze poprzez  $R_2$ , do sumy tego rezystora wraz z oporem wytwarzanym poprzez czujnik wygięcia  $R_1$  - który

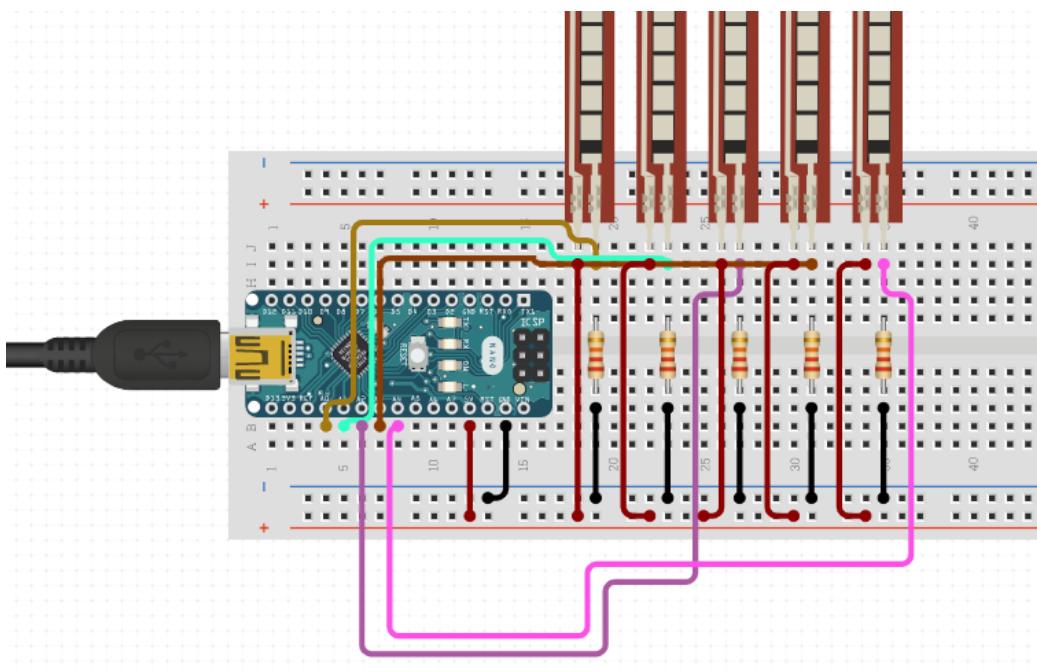
jak opisano w podrozdziale 6.1.2 jest zmienny. Oznacza to że im bardziej czujnik wygięcia jest zgięty, wytwarza on większy opór a co za tym idzie napięcie wyjściowe spada. Miara ta obrazuje jak bardzo palec jest zgięty i jest możliwa do uzyskania właśnie dzięki zastosowaniu układu dzielnika napięcia [1].

## 6.2 Budowa rękawicy

W poniższej sekcji zaprezentowano sposób w jaki zostały złączone wszystkie elementy rękawicy wspomniane w sekcji 6.1, aby była gotowa na oprogramowanie mikrokontrolera tworząc finałowy produkt. W tym celu została wybrana rękawica budowlana o grubych niciach ze ściągaczem wokół nadgarstka w celu zapewnienia komfortu, jak i precyzji położenia. Wybór ten również jest uzasadniony faktem początkowego planu dotyczącego wszycia materiału bezpośrednio w ręk-

---

wice jak i elastyczności które zapewnia grubsza rękawica. Tak jak wspomniano w podsekcji 6.1.2, do połączenia elementów została wykorzystana nić przewodząca. Dzięki grubym włóknom odstępu pomiędzy nićmi przewodzącymi prąd mogły być mniejsze, bez obawy przed spięciami w trakcie poruszania ręką. Dla tego projektu kontroler jest budowany dla lewej dłoni. Układ przewodów kontrolera jest zobrazowany na rysunku 6.5. Jest to jedynie obraz poglądowy, przedstawiony na płytce prototypowej a szczegółowy opis połączeń kontrolera zostanie opisany poniżej.



Rysunek 6.5: Poglądowy układ kontrolera przedstawiający sposób podłączenia dzielnika napięcia

Źródło: <https://www.circuito.io/app?components=514,8606,8606,8606,11022>

Mikroprocesor został umieszczony na wysokości centralnej części dłoni, przy kciuku, skierowany złączem USB na zewnątrz, pozwalając na łatwe podłączenie przewodu zasilającego jak i również ustawienie wejść analogowych w stronę



Rysunek 6.6: Efekt końcowy rękawicy-kontrolera

palców lewej dłoni. Z tej strony płytki Arduino będziemy też korzystać - zarówno przy wyprowadzaniu napięcia jak i uziemienia. Montaż samej płytki nie stanowi żadnego problemu ze względu na cztery otwory w rogach pozwalające na mocowanie, do którego została użyta zwykła nić do szycia. Aby stworzyć obwód wychodzący od pinu GND (z ang. Ground) czyli właśnie uziemienia, należy najpierw poprowadzić go przez rezystory. W tym celu wyjścia oporników zostały wygięte w pętle oraz zalutowane aby w łatwy sposób można było je przyszyć do rękawicy. Aby dodatkowo ułatwić sobie to zadanie - dodatkowo zostały one przyklejone bezpośrednio do materiału na bardzo małą ilość kleju. Umiej-

---

scowienie rezystorów jest po zewnętrznej wierzchniej stronie dloni, zaczynając się na wysokości kontrolera, zmierzając szeregowo w stronę nadgarstka. W ten oto sposób nić przewodząca została poprowadzona na zewnętrzną część dloni a następnie poprzez wszystkie pętle rezystorów, kończąc na ostatnim. Z drugiej strony rezystora dochodzi natomiast nić pochodząca z czujnika oporu. W tym celu podczas tworzenia czujników oporu pozostawiono 25 cm długości nici, co pozwoliło na przy mocowanie sensorów, a następnie połączenie obwodu. Sensory zostały zaopatrzone w krótkie, 1-2 cm długości kawałki taśmy elastycznej która została przyklejona na czubku każdego z sensorów pozwalając na elastyczny ruch sensora bez zerwania nici. Taśma ta została przyszyta w czubkach palców co dało punkt zaczepu dla sensorów. W tym momencie pozwoliło to na wygodne używanie nici wychodzących z sensorów w celu zamknięcia obwodu. Z powodu dużego zagęszczenia nici, które nie mogą się ze sobą stykać podczas ruchu ręki, ważnym dla projektu było naprzemienne używanie przestrzeni na rękawicy od spodu jak i od góry. Dzięki temu nici uziemienia oraz napięcia mogą się krzyżować nie zaburzając przy tym odczytów z sensorów. W celu jak najlepszego zagospodarowania przestrzenią wokół sensorów, uziemienie kciuka oraz małego palca zostały umiejscowione na nicie wychodzącej z prawej strony sensora, natomiast napięcie po lewej stronie. Dla pozostałych palców ustawienie to jest odwrotne. Mając to na uwadze została poprowadzona nić od kciuka poprzez wyjście 3.3 V na płytce a następnie wzduż krykci rękawicy, pozwalając nicią odpowiedzialnym za napięcie w odpowiednich sensorach na przyczepienie się w celu poboru napięcia tworząc tym samym dodatnią stronę układu. Nici wychodzące z sensorów które natomiast nie zostały do tej pory użyte, zostały przeszyte poprzez wyjścia analogowe a następnie podłączone kolejno do odpowiadających im rezystorów. Wyjścia odpowiadające każdemu z palców zostały opisane w tabeli poniżej.

---

Palec	Wyjście analogowe
Kciuk	A3
Wskazujący	A0
Środkowy	A1
Serdeczny	A6
Mały	A7

Sposób w jaki zostały dobrane wyjścia jest podyktowany zaleceniami o nie korzystaniu z wyjść analogowych A4 oraz A5 oraz pozostawieniu przestrzeni wokół wyjścia A3 przeznaczonego na kciuka, ponieważ jako jedynie połączenie musiało najpierw zmierzać w kierunku palców a następnie w dół w celu połączenia z rezystorem. Efekt końcowy pracy przedstawia zdjęcie 6.6.

### 6.3 Oprogramowanie mikrokontrolera

Do tej pory została opisana teoria omawianego kontrolera, elementy które zostały użyte w projekcie a także sposób ich połączenia. W niniejszej sekcji przedstawiony jest kod programu który został napisany w środowisku programistycznym oraz języku o tej samej nazwie - Arduino. Język ten poza drobnymi zmianami opiera się na języku C/C++, a jego pełny kod źródłowy jest dostępny na platformie Github [7]. Aby rozpocząć pracę z wybranym produktem od Arduino należy wykonać pewne czynności przygotowawcze, takie jak instalacja sterowników płytka dla danego systemu czy sposób obsługi w samym środowisku Arduino. Czynności te zostały opisane na stronie producenta, w związku z czym nie zostaną one szczegółowo opisane [28]. Po spełnieniu wszystkich wymagań, został przygotowany program który został napisany oraz wgrany do mikrokontrolera, który można podzielić na trzy części:

- Deklaracje
- Ustawienie mikrokontrolera

- 
- Główna pętla programu

których cel oraz opis został przedstawiony poniżej.

W pierwszej kolejności zostanie przedstawiona część deklaracji, w której to dołączono wymagane biblioteki dla poprawnej obsługi wszystkich sensorów. Pierwszą biblioteką która została dołączona do programu jest *Arduino\_LSM9DS1.h*. Biblioteka ta jest odpowiedzialna za obsługę IMU która przekazuje dane poprzez I2C do mikroprocesora. Biblioteka ta zajmuje się obsługą połączenia jak i kalibracja całego modułu. Inicjalizacja wartości poprzez bibliotekę wygląda w następujący sposób [26]

Sensor	Zakres	Częstotliwość
Akcelerometr	$[-4, +4]g - / + 0.122mg$	104Hz
Żyroskop	$[-2000, +2000]dps + / - 70mdps$	104Hz
Magnetometr	$[-400, +400]uT + / - 0.014uT$	20Hz

Tak jak wspomniano w sekcji 6.1.1 do połączenia bezprzewodowego Arduino Nano 33 BLE wykorzystuje moduł Bluetooth Low Energy, w związku z czym w części deklaracji dołączamy przeznaczoną do tego bibliotekę o nazwie *ArduinoBLE.h*. Biblioteka ta pozwala na dostęp oraz sterowanie modułem BLE ( z ang. Bluetooth Low Energy), i na potrzeby projektu zostaną opisane używane elementy z poniższych klasy których znajomość jest wymagana w celu zrozumienia napisanego programu. Klasy te to

- **BLEService**
- **BLECharacteristic**
- **BLEDescriptor**
- **BLE**
- **BLEDevice**

---

Szczegóły na temat zasad działania BLE są podane w sekcji ??, w tej sekcji jedynie zostanie opisana struktura tych klas. *BLEService* umożliwia nawiązanie połączenia z innym urządzeniem obsługującym bluetooth i jako parametr przyjmuje UUID - jest to jedyny paramater jaki należy podać i w ten sposób zostaje stworzony nowy serwis BLE dostępny pod tym identyfikatorem. Klasa *BLECharacteristic* tworzy nową cechę którą należy przypisać do danego serwisu. Cechą ta może zostać zadeklarowana jako cecha wybranego z pośród dostępnych typów w Arduino bądź jako uniwersalna poprzez podstawową klasę *BLECharacteristic*. Klasa ta przyjmuje trzy wymagane parametry: UUID, właściwości cechy oraz wartość. Wartość może zostać zadeklarowana jako podany ciąg znaków, bądź też poprzez określenie rozmiaru danych i wartość początkową. Wartość początkowa nie musi być podana w trakcie deklaracji. Ważnym elementem tej klasy jest określenie jej właściwości. W zależności od przeznaczenia mamy do wybory następujące opcje, które mogą być ze sobą łączone: BLEBroadcast, BLERead, BLEWriteWithoutResponse, BLEWrite, BLENotify, BLEIndicate. W celu wysyłania danych podczas gdy dane te się zmienią cechy w prezentowanym programie przyjęły dwie wartości BLE: read ( z ang. czytać) oraz notify (z ang. powiadomić). Klasa *BLEDescriptor* jest niejako klasą pomocniczą. Wartości tej klasy przypisuję się do danej cechy w celu lepszej obsługi serwisy oraz łatwiejszego rozpoznawania w przypadku pracy ze skomplikowanymi serwisami. Standardowo należy podać UUID danego deskryptora jako parametr, a także jego wartość jako ciąg znaków, bądź wartość w postaci tablicy bajtów oraz maksymalnego rozmiaru danych. Są to podstawowe klasy znajdujące się w części deklaracji których użycie zostały przedstawione na listingu 6.1 prezentującym wycinek części deklaracji [27]. Pozostałe dwie klasy zostaną opisane w części głównej pętli programu gdzie zostanie również pokazane ich zastosowanie.

```

1 #include <Arduino_LSM9DS1.h>
2 #include <ArduinoBLE.h>
3 BLEService imuService("1101");
4
5 BLECharacteristic imuAccChar("2101", BLERead | BLENotify,
6 , 12);
7 BLECharacteristic imuGyroChar("2102", BLERead | BLENotify,
8 , 12);
9 BLECharacteristic fingersChar("2103", BLERead | BLENotify,
10 , 20);
11
12 BLEDescriptor     imuAccDescriptor("3101", (byte*) "Acc
13 Descriptor", 15);
14 BLEDescriptor     imuGyroDescriptor("3102", (byte*) "Gyr
15 Descriptor", 15);
16 BLEDescriptor     fingersDescriptor("3103", (byte*) "Fin
Descriptor", 15);

```

Listing 6.1: Część deklaracji programu mikrokontrolera

Następną częścią programu jest ustawienie mikrokontrolera oraz sprawdzenia poprawności działania sensorów. Gdy proces ten zakończy się powodzeniem zostanie uruchomiona główna część programu - funkcja *loop()*, która wykonuje się nieprzerwania pozwalając kontrolować pracę naszego kontrolera. Za część związaną z inicjalizacją jest odpowiedzialna funkcja *setup()*. Funkcja ta jest wywoływana tylko raz za każdym razem gdy płytka jest podłączona do prądu bądź zostanie zresetowana [3]. W funkcji tej wywoływana jest statyczna metoda *begin(z ang. rozpoczęć)* na klasie *IMU* która należy do biblioteki *Arduino\_LSM9DS1.h*, i to właśnie ta funkcja pozwala na inicjalizację wspominanych sensorów wchodzących w skład jednostki pomiarowej - program zapętli się w tym miejscu jeśli funkcja ta zwróci błąd, poprzez rozpoczęcie pętli z warunkiem zawsze prawdziwym, co przedstawia listing 6.2 [26].

```

12 if (!IMU.begin()) {
13     Serial.println("Failed to initialize IMU!");
14     while (1);
15 }

```

```
16  
17     if (!BLE.begin()) {  
18         Serial.println("Failed to start Bluetooth!");  
19         while (1);  
20     }
```

Listing 6.2: Inicjalizacja IMU oraz BLE

Jeżeli sensory działają poprawnie, sprawdzana jest możliwość połączenia poprzez moduł bezprzewodowy. W tym celu wykorzystywana jest statyczna klasa *BLE*, która odpowiada za włączenie modułu Bluetooth oraz jego ustawienia. Tak jak w przypadku klasy *IMU*, program zapętli się w tym miejscu zwracając błąd jeżeli połączenie nie będzie możliwe. W przypadku powodzenia program rozpoczyna łączenie zadeklarowanych elementów, takich jak serwis, cechy oraz deskryptory tych cech. Klasa *BLE* posiada funkcje pozwalającą ustawić nazwę dla naszego połączenia, jej adres a przede wszystkim gdy wszystkie elementy są już gotowe wywołuję metodę *advertise()*, która pozwala na połączenie się z naszym serwisem. Na listingu 6.3 pokazane jest szczegółowa konfiguracja klasy BLE oraz serwisu a także odpowiednie użycie funkcji dla jednej z cech, które należy powtórzyć dla każdej dodawanej cechy z odpowiednimi wartościami.

```
21     BLE.setLocalName ("VrGlove");  
22     BLE.setAdvertisedService(imuService);  
23  
24     imuService.addCharacteristic(imuGyroChar);  
25     imuGyroChar.addDescriptor(imuGyroDescriptor);  
26     imuGyroChar.writeValue((byte) 0x01);  
27  
28     BLE.addService(imuService);  
29     BLE.advertise();
```

Listing 6.3: Obsługa serwisu przy użyciu klasy BLE

W ten oto sposób kończy się funkcja *setup()* i zostaje wywołana kolejna funkcja która od tej pory będzie odpowiedzialna za pracę kontrolera - *loop()*. W funkcji tej zostaje zadeklarowana ostatnia z wymienionych klas do obsługi Bluetooth. Klasa

---

ta nazywa się *BLEDevice* i jest bezpośrednio powiązana z urządzeniem które jest aktualnie podłączone. Dopóki nie ma podłączonego do kontrolera urządzenia, żadne czynności nie zostają podjęte. W momencie uzyskania danych z modułu łączności o nawiązanym połączeniu zostaje uruchomiona pętla, funkcjonująca tak długo jak połączenie to nie zostanie przerwane. Fragment kodu znajduje się na listingu 6.4 [27]. Niestety w trakcie pracy z urządzeniem odkryto błąd związany z rozłączeniem się centrali. Problem pojawi się gdy urządzenie zostaje rozłączone z mikrokontrolerem - w tym momencie klasa *BLE* nie zawsze zgłasza informacje o rozłączeniu, myśląc że urządzenia cały czas są podłączone. Problem został zauważony, jednak nie jest rozwiązany na stan z maja 2020 roku [24].

```
30     BLEDevice central = BLE.central();
31     if (central) {
32         Serial.print("Connected to central: ");
33         Serial.println(central.address());
34     }
35     while( central.connected()) {
36         [...]
37     }
```

Listing 6.4: Oczekiwanie na połączenie z urządzeniem przez mikrokontroler

Gdy wszystkie dotychczasowo opisane elementy programu nie zgłoszą problemów, następuje ostatnia faza którą jest zbieranie i przesyłanie danych. W pierwszej kolejności sprawdzany jest warunek czy jednostka pomiarowa ma dostęp do nowych danych. W przypadku tej aplikacji dane które zostały poddane analizie pochodzą z żyroskopu oraz z akcelerometru i są wyrażone jako zmienne *x,y* oraz z typu *float*. Z powodu naturalnych drgań dloni dane te ciągle się zmieniają w mikro skali która nie ma wpływu na efekty, niemniej jednak możemy założyć że gdy mikroprocesor nie jest przymocowany do stałego obiektu, dane te będą dostarczane z częstotliwością pracy sensorów. Za odczyt danych z żyroskopu i akcelerometru odpowiadają odpowiednio funkcje *readGyroscope(x,y,z)* oraz *re-*

---

*adAcceleration(x,y,z)* z klasy *IMU* które przypisuję odpowiednie wartości do podanych parametrów. Ważnym elementem podczas odczytu tych danych jest tak zwany szum który powstaje w trakcie przygotowania sensorów. W trakcie pierwszych odczytów szum ten został zmierzony oraz usunięty z pomiarów. Akcelerometr położony na płaskim obiekcie prawidłowo podawał odczyty, czyli zwracał wektor  $[0, 0, g]$  gdzie  $g$  oznacza grawitację ziemską. Żyroskop natomiast wskazywał błąd rzędu  $[2.80, 0.18, 0.18]$  w związku z czym od każdego odczytu właśnie taką wartość należy odjąć w celu uzyskania prawidłowych danych - czyli wektora zbliżonego do  $[0, 0, 0]$  w pozycji w której został skalibrowany. Jak wspomniano w sekcji ?? dotyczącej żyroskopu, wartości zwracane są podane w  $\frac{rad}{s}$ , również znane jako dps (z ang. Degrees per second) czyli w kątach na sekundę. W celu określenie kątów w jakim urządzenie się znajduje w danym momencie musimy te dane przemnożyć przez częstotliwość z jaką są one pobierane - czyli przemnożyć przez czas co zwróci nam miarę kątów. Osiągamy to poprzez zapisanie czasu w którym dane zostały pobrane a także poprzez zapisanie tej wartości przed następnym pobraniem danych jako czas ostatniego poboru. Różnica pomiędzy tymi wartościami daje czas jaki upłynął aby uzyskać nowe wartości. Zaczynając w pozycji kalibracyjnej, z idealnie usuniętym szumem, żyroskop zwróci wartość zero, a wraz ze zmianą wartości żyroskopu, suma tych zmian wskaże na orientację kontrolera względem pozycji wyjściowej. Więcej informacji na ten temat znajduje się w rozdziale ?? [9]. Ostatnią cechą którą chcemy przekazać są dane pobierane z sensorów wygięcia w celu określenia pozycji palców. Tak jak wspomniano w sekcji 6.1 dane te pobieramy przez mierzenie napięcia jakie znajduje się na poszczególny analogowych pinach wejściowych. Odczyty te uzyskujemy poprzez wywołanie metody *analogRead(pin)* [3]. Pobieranie danych oraz ich wstępna obróbka przedstawia listing 6.5.

---

```

38 if (IMU.accelerationAvailable() && IMU.gyroscopeAvailable
39   ()) {
40     previousTime = currentTime;
41     currentTime = millis();
42     elapsedTime = (currentTime - previousTime) / 1000;
43
44     IMU.readAcceleration(acc[0], acc[1], acc[2]);
45     IMU.readGyroscope(gyro[0], gyro[1], gyro[2]);
46
47     gyro[0] -= 2.80;
48     gyro[1] -= 0.18;
49     gyro[2] -= 0.18;
50
51     fingers[0] = analogRead(A3);
52     [...]
}

```

Listing 6.5: Wczytywanie danych z sensorów.

Na tym etapie mamy już dostęp do danych z akcelerometru w  $\frac{m}{s^2}$  oraz dane z żyroskopu wyrażone w stopniach. Informacje te mogłyby zostać przesłane w takiej formie, jednak z racji znajomości projektu, dane można dodatkowo skorygować poprzez zastosowanie dodatkowych filtrów. Podstawowym zabiegiem który został zastosowany jest eliminacja danych nieznaczących, czyli szumu który powstaje poprzez naturalne drgania ciała. Aby to osiągnąć, w programie przechowywane są poprzednie wartości z poszczególnych sensorów. Jak dane znaczące dla akcelerometru przyjęto różnice  $\pm 0.1$ , a dla danych z sensorów wygięcia  $\pm 15$ . Dla żyroskopu natomiast przyjęto wartości różniące się o przynajmniej  $\pm 0.2$ , a także dodatkowo sprawdzane jest czy dane z żyroskopu nie są w pozycji przy kalibracyjnej. Oznacza to że jeżeli wartości zwracane z żyroskopu wynoszą  $0 \pm 0.1$ , zostaną one zamienione na wartość równą 0. Filtrowanie wartości pokazane jest na listingu 6.6. W ten oto sposób otrzymano wartości z sensorów które były gotowe do wysłania na inne urządzenie. Wartości te jednak są wyrażone w postaci tablic typu *float*, natomiast jako wartości cechy bluetooth przyjmują tablice by-

---

tów. W związku z tym przed przypisaniem danych są przekazywane adresy tablic do nowych zmiennych, a następnie zmienne te zapisane w poszczególnej cęsie przy użyciu metody *setValue(value,valueSize)*. Rozmiar jest ten określony jako 12 ponieważ mamy do czynienia z tablicą 3 zmiennych typu *float* z których każda zajmuje 4 bajty pamięci. Zapisywanie danych jest pokazana na listingu 6.7 dla jednej cechy - proces ten należy powtórzyć dla wszystkich danych. Program następnie zapisuje obecne dane jako dane obecnej pętli, tak aby następne odczyty mogły zostać porównane z obecną iteracją, tym samym kończąc wykonywaną pętle. Warto zauważyć że program pomimo swojego głównego działa w funkcji *loop()*, jest wykonywany w ramach jednej iteracji jak tylko dojdzie do połączenia kontrolera z odbiornikiem.

```
53 | for (int i = 0;i<3;i++) {
54 |     if(!(gyro[i] < oldGyroData[i]-0.2 || gyro[i] >
55 |         oldGyroData[i]+0.2)){
56 |         if( gyro[i] < -0.1 || gyro[i] > 0.1){
57 |             gyro[i] = oldGyroData[i];
58 |         }else{
59 |             gyro[i] = 0.0;
60 |         }
61 |
62 |         if(!(acc[i] < oldAccData[i]-0.02 || acc[i] > oldAccData
63 |             [i]+0.02)){
64 |             acc[i] = oldAccData[i];
65 |         }
66 |
67 |     for (int i = 0;i<5;i++){
68 |         if(!(fingers[i] < oldFingersData[i]-15.0 || fingers[i] >
69 |             oldFingersData[i]+15.0)){
70 |                 fingers[i] = oldFingersData[i];
71 |             }
72 |     }
```

Listing 6.6: Wstępne filtrowanie danych z sensorów.

```
73 byte *accChar = (byte*)&acc;  
74 imuAccChar.setValue(accChar, 12);
```

Listing 6.7: Zapisywanie danych do cech w serwisie.

## 6.4 Dalszy rozwój

W rozdziałach ?? i ?? pokazano budowę kontrolera oraz aplikacji która wykorzystuje zbudowany kontroler w celu obsługi podstawowych funkcji rękawicy-kontrolera. Projekt ten powstał z myślą ograniczonego budżetu, prostoty wykonania oraz możliwości replikacji. Założenia te spowodowały że zdecydowano się na pewne rozwiązania które w końcowej wersji projektu pokazały swoje wady. W tym rozdziale zostanie poruszony temat błędów popełnionych w pierwszej wersji tego projektu oraz przykładowe sposoby na ich rozwiązanie w przyszłości.

Przypis z pracy dla wszystkich podrozdziałów

Wykorzystania projektu pokazano w drugiej pracy

### 6.4.1 Problemy mikroprocesora

Przede wszystkim szukano małego mikrokontrolera tak aby nie był on przeszkodą podczas użytkowania kontrolera. O ile założenie to było dobrym pomysłem, okazało się że umiejscowienie przy brzegu sprawiło że ruch palców, w szczególności kciuka, może zmienić położenie jednostki IMU na rękawicy. Oznacza to że nawet jeśli nasza ręka znajduje się w stałej pozycji, samo poruszanie palcami wprowadza błąd w odczycie. Niestety wybór tego produktu od Arduino również przysporzył wiele kłopotów z racji błędnego rozłączania się adaptera Bluetooth. Z dotychczasowego użytkowania można stwierdzić iż kontroler poprawnie łączy się i rozłącza dwa razy, zaś w większości testowanych przypadków dochodzi do błędu połączenia przy trzeciej próbie. Aby usunąć ten błąd należy odłączyć płytę od zasilania i podłączyć ponownie, resetując tym samym moduł. Samo oprogramowanie rękawicy skupia się na odczytach z dwóch sensorów. Praktyka pokazała że dane te nie

---

są wystarczająco dokładne, i jeżeli to możliwe powinny być pobierane również dane magnetometru w celu dodatkowego korygowania odczytów z żyroskopu.

#### **6.4.2 Problemy budowy i czujników wygięcia**

Problem z użytkowaniem rękawicy pojawił się dość szybko od jej zbudowania. Mianowicie wybrana do projektu rękawica była zbyt gruba, powodując dyskomfort w użytkowaniu w szczególności przez dłuższy okres czasu. Początkowe kryterium elastyczności, przesądziło o wybraniu tej rękawicy, jednak cienka rękawica również spełniła by wymagania końcowego produktu. Rozwiążanie zastosowane w celu odczytu wygięcia palców z założenia wyglądało na idealnie pasujące do wymagań projektu. Pomimo swojej prostoty wykonania oraz braku pomiaru takich cech jak odwodzenie palców czy też wygięcie poszczególnych stawów, spełnia ono swoją podstawową funkcję. Problemem tego rozwiązania jest natomiast brak elastyczności sensorów. W momencie zgięcia palców droga od knykci do paznokci się wydłuża sprawiając że sensor jest poddawany sile nacisku od strony palca która jest tym spowodowana. Sensory te pomimo braku elastyczności są zbudowane z materiału wytrzymały na rozciąganie dzięki czemu nie pękają podczas zgięcia palców, jednak w celu zapewnienia lepszego mocowania i większej ochrony, z jednej strony została przymocowana elastyczna guma która trzyma sensor przy czubkach palców, z drugiej natomiast sensor został wszyty w rękawicę. Problem który się pojawił w trakcie użytkowania pochodził ze sposobu wszycia sensora. Została do tego użyta nić przewodząca która z powodu rozciągliwości rękawicy nie mogła zostać wszyta na sztywno, w związku z czym stawała ona mniejszy opór podczas zginania palców i niejako została wyciągnięta przez sensor, powodując tym brak dokładności odczytów. Nić ta oprócz niskiej elastyczności okazała się być nietrwała. W trakcie korzystania z rękawicy doszło do kilku pęknięć, które zostały ponownie związane, jednak została przerwana w ten

---

sposób ciągłość obwodu. Przez dodanie dodatkowych wiązań odczyty z sensorów się pogorszyły, sprawiając że wygięcie palca wskazującego ma większy wpływ na odczyty z kciuka, niż zgięcie kciuka samo w sobie. Podobna sytuacja przytrafiła się z sensorem małego palca oraz serdecznego. Mała powierzchnia na dłoni wokół której należało poprawić wiele połączeń, sprawiła że część nici była blisko siebie, powodując momentami odczuwalne mrowienie na dłoni. Problem ten został rozwiązyany poprzez zastosowanie izolacji od wewnętrznej strony rękawicy, jednak nie gwarantuje to przeciwdziałaniu zwarć w obwodzie. Konkludując, nić przewodząca nie jest najlepszym rozwiązaniem w celu połączenia elementów dla tego projektu i powinno zostać zastąpiona trwałszym połączeniem. Gdyby jednak została ona użyta, element przewodzący powinien znajdować się w środku oplotu, bądź powinny zostać zastosowane inne sposoby izolacji, a sama wytrzymałość nici powinna być znacznie większa. Mocowanie sensorów wygięcia powinno być bardziej trwałe oraz statyczne, nie pozwalając na przemieszczenie sensora na palcu. Alternatywą dla tego rozwiązania jest wykorzystanie czujników pomiaru wygięcia opartych o światło nadawane z jednej strony plastikowej tuby oraz miernika natężenia światła z drugiej. W ten sposób wiadomo że im mniejszy pomiar otrzymywanej światła, tym większe wygięcie tuby, której załamanie blokuje bezproblemowy dopływ światła. Rozwiązanie to również zapewnia pomiary nieznieksztalcone poprzez zachowanie innych sensorów a także wygląda na bardziej dokładne [19].

#### **6.4.3 Animacja modelu**

Ostatnim elementem aplikacji dla projektu jest zapewnienie animacji dloni. W tym celu został wykorzystany *Google Sceniform*, dzięki któremu zimportowano modele, ustawiono scenę, przypisano model a także obsługiwano przemieszczenie i orientację. Ostatnim brakującym elementem jest animacja modelu. Według

---

dokumentacji starszej wersji projektu osiągnąć to można poprzez klasę *Skeleton-Node*, pozwalającą na dostęp do kości modelu, bez wykorzystania zewnętrznego programu graficznego. Jednak z niejasnych przyczyn klasa ta została usunięta w ostatniej wersji SDK, powodując brak możliwości wprowadzania zmian w modelu który został zaimportowany przy użyciu wtyczki. Problem ten rozwiązano poprzez wykorzystanie programu *Blender*, dzięki któremu można było wyeksportować modele w wyznaczonej pozycji. Aby osiągnąć jednak animację modelu w czasie rzeczywistym, na podstawie dostępnych danych z sensorów wygięcia - cała klasa renderująca fragment ?? musi zostać napisana od nowa z wykorzystaniem innej technologii, ponieważ na oficjalnej stronie dystrybucji *Sceneform*, jest napisane iż projekt został zarchiwizowany, w związku z czym taka opcja nie zostanie dodana [14].

#### 6.4.4 Błąd rotacji

W przypadku rotacji jest wiele sposobów na polepszenie rezultatów. W prezentowanym projekcie wybrano podstawową metodę która wykorzystuje jedynie żyroskop oraz akcelerometr i przy ich użyciu wykorzystuje filtr komplementarny. Tak jak wcześniej wspomniano, aby dokładnie skorygować żyroskop na wszystkich trzech osiach, należy wykorzystać również magnetometr. Oprócz tego istnieje wiele filtrów takich jak Kalmana czy Madgwick'a które skutecznie usuwają szum, a także algorytmy wykorzystujące nowe pomiary w połączeniu z tymi zebranymi przed nimi. Możliwości łączenia technik udoskonalania odczytu rotacji z jednostek IMU sprawia, że nie ma jednego najlepszego rozwiązania, a ich wybór jest uzależniony on rodzaju projektu nad którym się pracuje [18].

---

#### **6.4.5 Problem obliczania przesunięcia**

Obliczenie położenia kontrolera w przestrzeni, niewątpliwie należy do najtrudniejszego problemu w tym projekcie. Sedno problemu tkwi w niedokładności danych. Z powodu wykorzystania metody podwójnego całkowania, błąd uzywany w cm przy pojedynczej całce, rośnie do m przy całkowaniu podwójnym. Ekran aplikacji jest mierzony w m, a ruch dloni z kontrolerem ma ograniczony zakres długości ramienia. Pomimo tego w niewielkim czasie błąd rośnie do poziomu w którym model znika z ekranu użytkownika. Niestety nie istnieje łatwy sposób na skorygowanie błędów powstałych w wyniku tego algorytmu. Firmy zajmujące się tym problemem dodatkowo umieszczają czujniki pozwalające określić odbiornik urządzenia i ustalić pozycję względem niego, kamery zewnętrzne obserwujące ruch w przestrzeni a także dodatkowe czujniki optyczne. W przypadku rękawicy-kontrolera który może poruszać się we wszystkich kierunkach dodatkowy problem stanowi rotacja, przez którą błąd staje się coraz większy. W przypadku prostej aplikacji nie wykorzystującej zaawansowanych jednostek pomiarowych oraz algorytmów filtrujących, często efekt jaki można osiągnąć tą techniką nie sprawdza się w zastosowaniu, dlatego też dla tej aplikacji domyślnie funkcja ta jest wyłączona [18].

### **6.5 Podsumowanie projektu**

W tym rozdziale podano informacje na temat projektu rękawicy kontrolera a także elementów które są wymagane w celu jego funkcjonowania. Zostały podane specyfikacje podzespołów, sposoby pozyskiwania danych, metoda komunikacji z innymi urządzeniami a także rozwiązano problemy związane z kosztem projektu. Krok po kroku przedstawiono złożenie elementów w celu uzyskania końcowej wersji produktu i ostatecznie omówiono zasadę działania programu napisanego

---

w Arduino wraz z bibliotekami które zostały użyte w ramach projektu, natomiast zasady ich praktycznego działania zostały pokazane na listingach. W ten oto sposób została zakończona główna część projektu, pozwalająca na wykorzystanie rękawicy kontrolera jako część większych przedsięwzięć. Aby jeszcze lepiej zobrazować sposób obsługi kontrolera, oraz metody wykorzystania danych, w rozdziale ?? zostanie pokazana przykładowa aplikacja wykorzystująca dane w celu analizy oraz prezentacji.

## **Rozdział 7**

### **Podsumowanie**

# Bibliografia

- [1] K. Academy. Voltage divider. <https://www.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider>, 7 2020.
- [2] Adafruit. Pressure-sensitive conductive sheet. <https://www.adafruit.com/product/1361>, 7 2020.
- [3] Arduino. Language reference. <https://www.arduino.cc/reference/en/>, 7 2020.
- [4] Botland.com. Arduino nano 33 ble - abx00030. <https://botland.com.pl/pl/arduino-seria-nano-oryginalne- plytki/14758-arduino-nano-33-ble-abx00030-7630049201491.html>, 7 2020.
- [5] Botland.com. Czujnik ugięcia 112 x 6,3 mm - sparkfun sen-08606. <https://botland.com.pl/pl/czujniki-nacisku/2406-czujnik-ugiecia-112-x-63-mm-sparkfun-sen-08606.html>, 7 2020.
- [6] BreqLabs. Exoglobe. <http://www.breqlabs.com/website/index.html>, 1 2019.

- 
- [7] P. Bugalski. (arduino, co w środku... – część 3 – źródło wbudowanych funkcji). <https://forbot.pl/blog/arduino-co-w-srodku-3-zrodlo-wbudowanych-funkcji-id17291>, 7 2020.
  - [8] Captoglove. Captoglove. <https://www.captoglove.com/>, 1 2019.
  - [9] Dejan. Arduino and mpu6050 accelerometer and gyroscope tutorial. <https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/>, 7 2020.
  - [10] J. D’Onfro. Facebook daje tę książkę wszystkim pracownikom działu wirtualnej rzeczywistości. <https://businessinsider.com.pl/technologie/nowe-technologie/ready-player-one-inspiracja-dla-oculusa/rh7ml9x>, 8 2016.
  - [11] Dreams. What happens in your brain while you dream? <https://www.dreams.co.uk/sleep-matters-club/what-happens-in-your-brain-while-you-dream/>, 7 2020.
  - [12] Forbot. Rezystor - co warto wiedzieć. <https://forbot.pl/blog/lekcykon/rezystor>, 7 2020.
  - [13] A. Gibbs. Movie-inspired inventions come to life. <https://www.cnbc.com/2015/01/14/back-to-the-future-inventions-how-movies-inspired-todays-technology.html>, 1 2015.
  - [14] Google. Sceneform overview. <https://developers.google.com/sceneform/develop>, 3 2020.
  - [15] GoTouchVr. Vrtouch. <https://www.gotouchvr.com/copy-of-technology-devices-1>, 1 2019.

- 
- [16] A. Jan. The evolution of gloves as input devices. <https://medium.com/@b00062378/the-evolution-of-gloves-as-input-devices-84efe6ff72bb>, 6 2019.
  - [17] E. B. V. P. Jérôme Perret. Touching virtual reality: a review of haptic gloves. *ResearchGate*, 2018.
  - [18] R. Labs. Um7 content library. <https://redshiftlabs.com.au/support-services/um7-content-library/>, 7 2020.
  - [19] A. D. Murray. Oprical flex sensors. <https://www.instructables.com/id/Optical-Flex-Sensor/>, 7 2017.
  - [20] P. J. K. Nancy Gupton. What's the difference between ar, vr, and mr? <https://www.fi.edu/difference-between-ar-vr-and-mr>, 6 2020.
  - [21] NeuroDigital. Avatarvr. <https://avatarvr.es/>, 1 2019.
  - [22] Noitom. Hi5. <https://hi5vrglove.com/>, 1 2019.
  - [23] S. Pwn. rzeczywistość. <https://sjp.pwn.pl/słowniki/rzeczywisto%C5%9B%C4%87.html>, 7 2020.
  - [24] Russell108. Ble nano 33 does not report or disconnect from central. <https://github.com/arduino-libraries/ArduinoBLE/issues/33>, 5 2020.
  - [25] P. Sarmini. Xr - rzeczywistość skrzyżowana, czy ukrzyżowana? <https://mu.pl/blog/xr--rzeczywistosc-skrzyzowana-czy-ukryzowana-13>, 7 2020.

- 
- [26] SM. Arduino lsm9ds3 library. <https://www.arduino.cc/en/Reference/ArduinoLSM9DS1>, 7 2020.
  - [27] SM. Arduinoble library. <https://www.arduino.cc/en/Reference/ArduinoBLE>, 7 2020.
  - [28] SM. Getting started with the arduino nano 33 ble. <https://www.arduino.cc/en/Guide/NANO33BLE>, 7 2020.
  - [29] L. D. Society. Lucid dream guides. <https://www.luciddreamsociety.com/how-to/>, 7 2020.
  - [30] Systel. Virtual reality, czyli czym jest wirtualna rzeczywistość? <https://systel.pl/virtual-reality/>, 7 2020.
  - [31] Tactai. Tactai haptic module. <https://www.tactai.com/>, 1 2019.
  - [32] Łukasz Michalik. Historia wirtualnej rzeczywistości. skąd wzięły się google vr? <https://gadzetomania.pl/57627,historia-vr>, 7 2020.
  - [33] Łukasz Michalik. Najlepsze filmy o wirtualnej rzeczywistości. jak wyobrażali ją sobie twórcy matriksa i 13. piętra? <https://gadzetomania.pl/57950,filmy-o-vr-wirtualna-rzeczywistosc>, 7 2020.
  - [34] M. VR. Manus prime. <https://manus-vr.com/>, 1 2019.

# **TODOs:**

■	IMU . . . . .	15
■	historia, pierwsze firmy, popularne rozwiązania . . . . .	15
■	proof read . . . . .	26
■	Opis: BLE, Rotacja, przesunięcie i animacja, przy opisywanych sensorach	26
■	Przypis z pracy dla wszystkich podrozdziałów . . . . .	46
■	Wykorzystania projektu pokazano w drugiej pracy . . . . .	46