



**POLSKO-JAPOŃSKA AKADEMIA
TECHNIK KOMPUTEROWYCH**

**WYDZIAŁ INFORMATYKI
KATEDRA SIECI KOMPUTEROWYCH
SIECI URZĄDZEŃ MOBILNYCH**

Kamil Kacprzak
s14004

**Rozwój technologii wirtualnej
rzeczywistości na przykładzie
rękawicy-kontrolera**

**Praca inżynierska
dr inż. Michał Tomaszewski**

Warszawa, Czerwiec, 2020

Spis treści

1	Wstęp	1
2	Zagadnienia	3
2.1	Czym jest wirtualna rzeczywistość?	3
2.2	Porównanie dostępnych środowisk do tworzenia aplikacji VR . . .	3
2.3	Porównanie dostępnych SDK	3
3	Komponenty rękawicy-kontrolera	4
3.1	Nawigacja inercyjna	4
3.2	IMU - Inercyjna jednostka pomiarowa	4
3.2.1	Żyroskop	4
3.2.2	Akcelerometr	4
3.2.3	Magnetometr	4
3.3	Stopnie swobody	4
3.4	Porównanie mikrokontrolerów	4
3.5	Monitorowanie położenia palców	4
4	Projekt rękawicy	5
4.1	Przegląd podzespołów i technologii użytej w projekcie	5
4.1.1	Mikrokontroler	6
4.1.2	Czujnik wygięcia	8
4.1.3	Rezystor	10
4.2	Budowa rękawicy	12
4.3	Oprogramowanie mikrokontrolera	15
4.4	Dalszy rozwój projektu	15
5	Dedykowana aplikacja w systemie Android	17
5.1	Interfejs	17
5.2	Komunikacja	17
5.2.1	Porównanie Bluetooth z Bluetooth Low Energy(BLE) . .	17
5.2.2	Pobieranie danych	17

5.3	Google Sceneform SDK	17
5.3.1	Model ręki	17
5.3.2	Rozpoznanie i animacja modelu	17
5.3.3	Dodawanie nowego modelu	17
5.3.4	Rotacja	17
5.3.5	Przesunięcie	17
5.4	Dalszy rozwój aplikacji	17
6	Podsumowanie	18
7	Zagadnienia wykorzystania szablonu	19
7.1	Structure	19
7.1.1	Top Matter	19
7.2	Minimalizacja parametrów funkcji plenoptycznej	19

Spis rysunków

4.1	Opis wejść oraz wyjść Arduino Nano 33 BLE z podręcznika producenta.	8
4.2	Sensor wygięcia własnego wykonania	10
4.3	Oznaczenia rezystorów	11
4.4	Układ dzielnika napięcia	12
4.5	Poglądowy układ kontrolera przedstawiający sposób podłączenia dzielnika napięcia	13
4.6	Efekt końcowy rękawicy-kontrolera	16

Streszczenie

Abstract

Rozdział 1

Wstęp

Poniższa praca przedstawia ewolucję rozwiązań oraz problemów z którymi mierzą się współczesne firmy, w celu stworzenia nowych, wydajnych a także intuicyjnych rozwiązań, w tym kontrolerów dla użytkowników świata wirtualnego. Kontrolery w szczególności napotykają wiele problemów ze względu na przeszkody wynikające z natury nawigacji inercyjnej, czyli określenia położenia i orientacji zazwyczaj w środowisku bez punktu odniesienia. Problemy te zostaną opisane w dalszej części tej pracy, a także metody jakie zostały zastosowane aby rozwiązać wspomniane problemy przez liderów branży.

Z biegiem czasu dąży się do jak najbardziej realistycznych rozwiązań, tak aby w pełni oddać naturę świata rzeczywistego w środowisku programowalnym. W tym celu powstały pierwsze komercyjne modele rękawic, pozwalających na ruch w przestrzeni świata wirtualnego, co dało możliwość naturalnego interfejsu pomiędzy użytkownikiem a światem w którym się znajduje. Naturalność tego rozwiązania, oraz potencjał jaki w sobie kryje, sprawiło że wiele firm zaczęło tworzyć własne rozwiązania tego produktu, dodając unikalne właściwości takie jak wibracje symulujące dotyk, punktową imitację nacisku czy też szkielet blokujący naszą dłoń. Rozwój tego rynku oraz chęć inwestycji w tego typu projekty ze strony dużych korporacji pokazuje zapotrzebowanie na tego typu rozwiązanie.

Potencjał który się w tym kryje, oraz brak uniwersalnego rozwiązania jest głównym powodem dla powstania tej pracy.

Celem tej pracy jest stworzenie własnego, uproszczonego kontrolera rękawicy, w celu lepszego poznania znanych problemów dla tego typu urządzeń oraz własnej implementacji ich rozwiązań. Kontroler ten powinien być prosty w wykonaniu, tak aby można było go odtworzyć w domowych warunkach, przy jednoczesnym spełnianiu podstawowych wymagań, które są od niego wymagane takie jak określenie orientacji, położenia oraz momentu zgięcia poszczególnych palców. Ostatecznie dążono do stworzenia aplikacji na system Android, która będzie łączyć się z kontrolerem oraz prezentować przesłane z rękawicy dane. Na podstawie tych informacji zostanie wyświetlony w czasie rzeczywistym model dłoni na ekranie smart-fona, który będzie reagował na zmiany w przesyłanych danych.

Słowa kluczowe: Wirtualna rzeczywistość, Kontroler, Nawigacja inercyjna, Akcelerometr, Żyroskop, Czujnik wygięcia

Rozdział 2

Zagadnienia

TODO:
ch:zagadnienia

- 2.1 Czym jest wirtualna rzeczywistość?**
- 2.2 Porównanie dostępnych środowisk do tworzenia aplikacji VR**
- 2.3 Porównanie dostępnych SDK**

Rozdział 3

Komponenty rękawicy-kontrolera

Opis rękawic na rynku

3.1 Nawigacja inercyjna

INS - inertial navigation system

3.2 IMU - Inercyjna jednostka pomiarowa

Jak dostępne rękawice omijają problem

3.2.1 Żyroskop

3.2.2 Akcelerometr

3.2.3 Magnetometr

3.3 Stopnie swobody

3.4 Porównanie mikrokontrolerów

3.5 Monitorowanie położenia palców

Rozdział 4

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. Po zaznajomieniu się z kodem aplikacji, będą zaprezentowane wady oraz możliwe ulepszenia projektu, jakie ukazały się w trakcie prac nad aplikacją obsługującą i wykorzystującą dane z rękawicy, w celu ich prezentacji o czym będzie więcej mowa w rozdziale 5 dotyczącym tejże właśnie aplikacji.

4.1 Przegląd podzespołów i technologii użytej w projekcie

Jak dowiedzieliśmy się z rozdziału 3 dotyczącego podstawowych komponentów rękawic, kluczowym dla powodzenia projektu jest ustalenie

Następny rozdział?
- Dodaj sekcje mówiący o wybraniu arduino ide oraz nie wybraniu istniejących sdk aby napisać własne połączenie BLE w javie bez unity

-
- orientacji
 - położenia
 - pozycji palców

W tym celu należy zebrać informacje z czujników, a następnie wszystkie te informacje należy przesłać 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 podrozdziałach 3.2.1 oraz 3.2.2. 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.

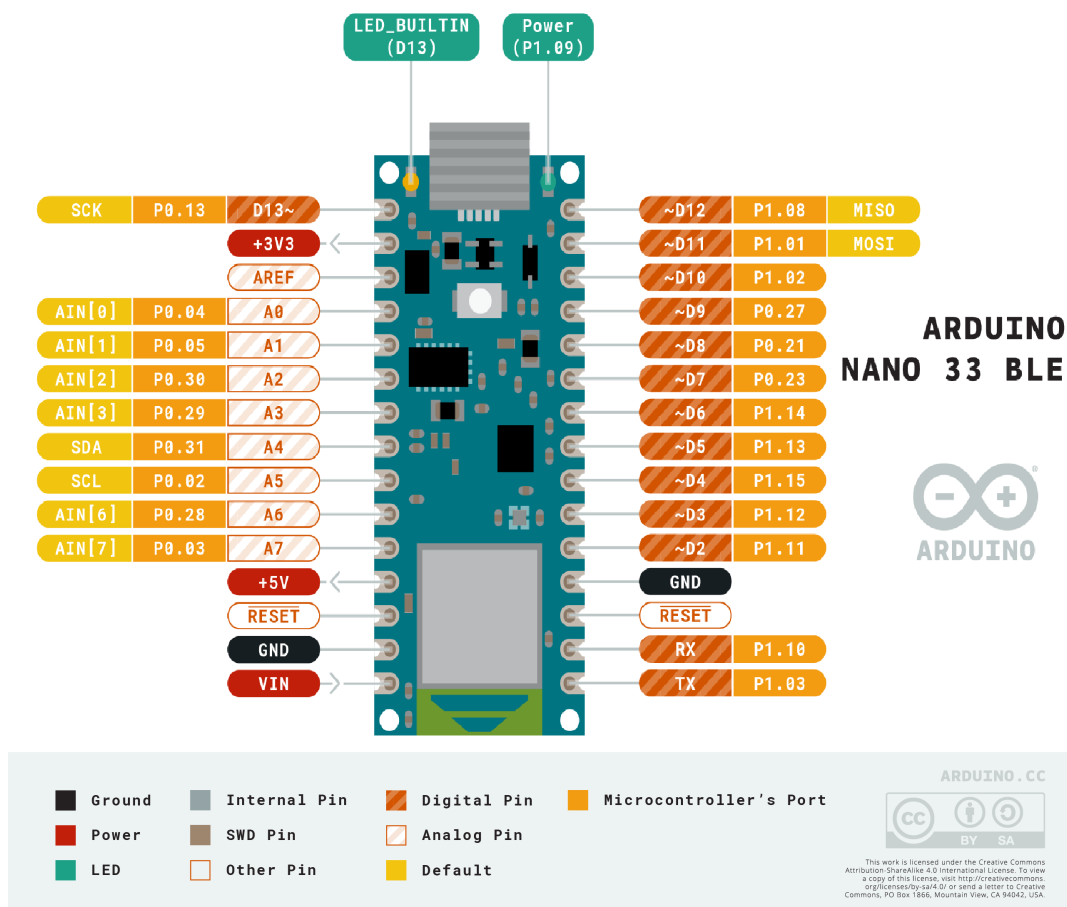
4.1.1 Mikrokontroler

Jak przed chwilą wspomniano, w projekcie wykorzystywana jest płytką od Arduino, która nosi nazwę Nano 33 BLE. Jest to małych rozmiarów płytką 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. Warto na wstępie zauważyć że Nano 33 BLE pracuje domyślnie wyłącznie z napięciem 3,3 V. Nie należy podłączać bezpośrednio zasilania o większym napięciu. W celu podłączenia zasilania 5 V należy zlutować zworę 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. Płytką 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 obsługiwanego 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. [3]

Z najważniejszych elementów układu płytki 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 związku z czym zalecane jest nie stosowanie tych wejść analogowych. Szczegółowy opis wejść/wyjść płytki przedstawiony jest na grafice 4.1.

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 4.2, a także uziemienie znajdujące się po tej samej stronie płytki.



Rysunek 4.1: Opis wejść oraz wyjść Arduino Nano 33 BLE z podręcznika producenta.

4.1.2 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 stronę 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 3 dotyczącym komponentów komercyjnych rękawic - większość producentów decyduje się na użycie wielu inercyjnych jednostek pomiarowych, na podstawie których są w stanie dokładnie określić położenia każdej części palca, bądź też specjalistycznych sensorów służących do pomiaru stopnia wygięcia czujnika względem pozycji prostej. Czujnik ten po podłączeniu do prądu zwiększa swój opór wraz ze zwiększonym 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 przystępne a jednocześnie pozwalające na osiągnięcie tego samego celu.

Dodaj jakieś źródło flex sensor albo imu na palcach

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 [4]. Jest ono stosunkowo proste w założeniach i wymaga jedynie dwóch kluczowych 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 przymocowania poszczególnych elementów przy jednoczesnym zapewnieniu funkcjonalności. Elementy te zostały sklejone na kawałku taśmy samoprzylepnej oraz dodatkowo sklejone przy brzegu aby nić nie wysliznęła się w trakcie korzystania z czujnika. Efekt końcowy jest widoczny na zdjęciu 4.2. 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 4.6.



Rysunek 4.2: Sensor wygięcia własnego wykonania

4.1.3 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 włą-

czony 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 sposobu 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 miliomów do megaomów. Spośród dostępnych rodzajów rezystorów w projekcie zostały użyte rezystory THT (z ang. Through-Hole Technology) - czyli tak zwane rezystory do montażu przewlekane. W tym rodzaju oporników rezystancja jest ilustrowana poprzez kolorowe paski umieszczone wokół oporu, co pozwala odczytać ich wartość według ilustracji 4.3. Alternatywą do tego sposobu jest podłączenie rezystora pod miernik elektryczny ustawiony w tryb pomiaru oporu [5].

Kolor	Wartość		Mnożnik	Tolerancja	Współczynnik temp.
	1 pasek	2 pasek		± %	± ppm/K
			3 pasek	4 pasek	ostatni pasek
brak	–	–	–	20	–
srebrny	–	–	10^{-2} (0,01 Ω)	10	–
złoty	–	–	10^{-1} (0,1 Ω)	5	–
czarny	–	0	10^0 (1 Ω)	–	250
brązowy	1	1	10^1 (10 Ω)	1	100
czerwony	2	2	10^2 (100 Ω)	2	50
pomarańczowy	3	3	10^3 (1 k Ω)	–	15
żółty	4	4	10^4 (10 k Ω)	–	25
zielony	5	5	10^5 (100 k Ω)	0,5	20
niebieski	6	6	10^6 (1 M Ω)	0,25	10
fioletowy	7	7	10^7 (10 M Ω)	0,1	5
szary	8	8	10^8 (100 M Ω)	0,05	2
biały	9	9	10^9 (1 G Ω)	–	–

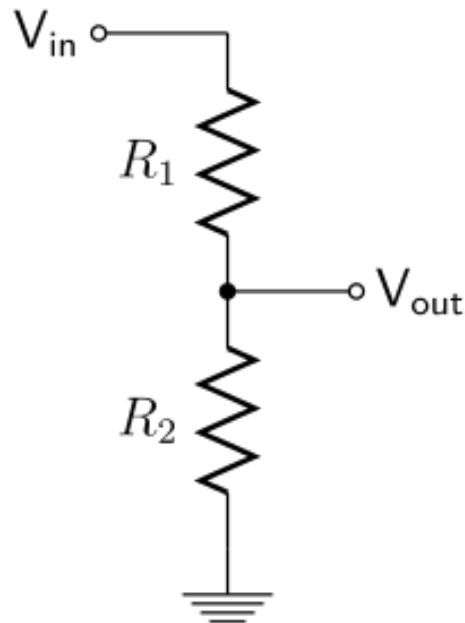
Rysunek 4.3: Oznaczenia rezystorów

Element ten jest kluczowy w celu ograniczenia przepływu prądu w obwodzie rękawicy, co pozwala na monitorowanie oporu wytwarzanego poprzez czujnik wygięcia. Sposób działania układu, nazywanego dzielnikiem napięcia jest pokazany na rysunku 4.4, oraz wyraża się wzorem

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

Wzór ten podaje napięcie wyjściowe ν_{out} , które równa się napięciu wejściowemu ν_{in} przeskalowanemu przez stosunek rezystorów. W opisywanym

przypadku jest to stosunek zastosowanego rezystora $4.7k\Omega$ wyrażonego we wzorze poprzez R_2 , do sumy tego rezystora wraz z oporem wytwarzanym poprzez czujnik wygięcia R_1 - który jak opisano w podrozdziale 4.1.2 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].

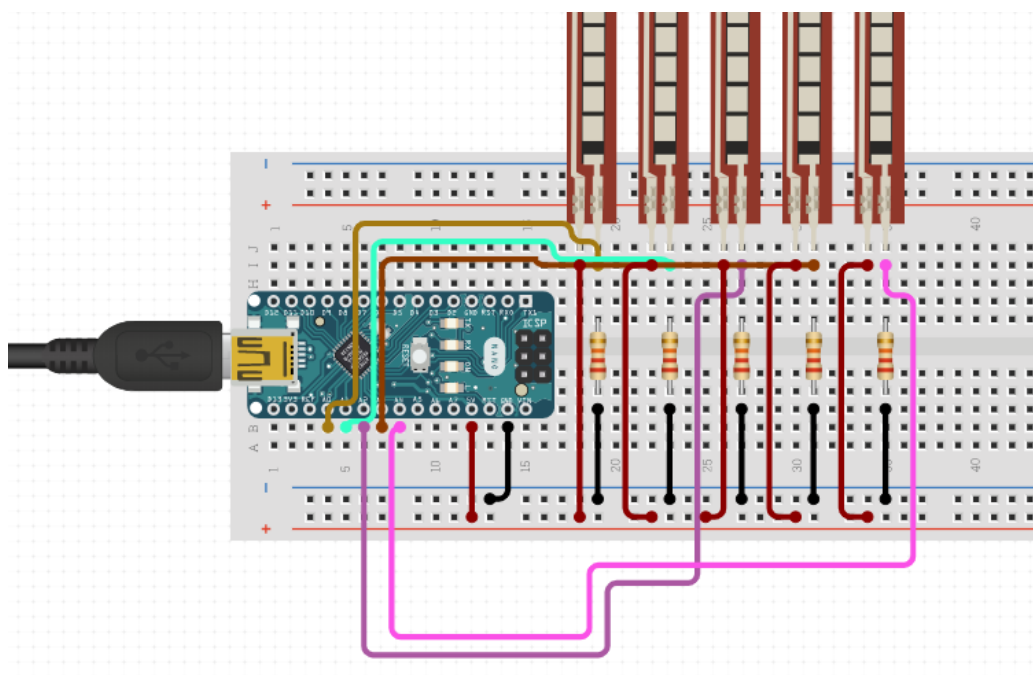


Rysunek 4.4: Układ dzielnika napięcia

4.2 Budowa rękawicy

W poniższej sekcji zaprezentowano sposób w jaki zostały złączone wszystkie elementy rękawicy wspomniane w sekcji 4.1, 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 zapew-

nienia komfortu, jak i precyzji położenia. Wybór ten również jest uzasadniony faktem początkowego planu dotyczącego wszycia materiału bezpośrednio w rękawice jak i elastyczności które zapewnia grubsza rękawica. Tak jak wspomniano w podsekcji 4.1.2, do połączenia elementów została wykorzystana nić przewodząca. Dzięki grubym włóknom odstęp 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 4.5. Jest to jedynie obraz poglądowy, przedstawiony na płytce prototypowej a szczegółowy opis połączeń kontrolera zostanie opisany poniżej.



Rysunek 4.5: Poglądowy układ kontrolera przedstawiający sposób podłączenia dzielnika napięcia

Mikroprocesor został umieszczony na wysokości centralnej części dłoni, przy kciuku, skierowany złączem USB na zewnątrz, pozwalając na łatwe podłącze-

nie przewodu zasilającego jak i również ustawienie wejść analogowych w stronę 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. Umieszczenie rezystorów jest po zewnętrznej wierzchniej stronie dłoni, zaczynając się na wysokości kontrolera, zmierzając szeregowo w stronę nadgarstka. W ten oto sposób nić przewodząca została poprowadzona na zewnętrzną część dłoni 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 przymocowanie 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 przestrzeni 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 wzdłuż knykcii rękawicy, pozwalając nicią odpowiedzialnym za napięcie w odpowiednich sensorach na przyłączenie 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 jedyne połączenie musiało najpierw zmierzać w kierunku palców a następnie w dół w celu połączenia z rezystorem. Efekt końcowy pracy przedstawia zdjęcie 4.6.

4.3 Oprogramowanie mikrokontrolera

4.4 Dalszy rozwój projektu



Rysunek 4.6: Efekt końcowy rękawicy-kontrolera

Rozdział 5

Dedykowana aplikacja w systemie Android

5.1 Interfejs

5.2 Komunikacja

5.2.1 Porównanie Bluetooth z Bluetooth Low Energy(BLE)

5.2.2 Pobieranie danych

5.3 Google Sceneform SDK

5.3.1 Model ręki

5.3.2 Rozpoznanie i animacja modelu

5.3.3 Dodawanie nowego modelu

5.3.4 Rotacja

5.3.5 Przesunięcie

5.4 Dalszy rozwój aplikacji

TODO:
ch:aplikacja

TODO: Krótki
opis o aplikacji i
rozdziale

Rozdział 6

Podsumowanie

TODO: Podsumowanie

Rozdział 7

Zagadnienia wykorzystania szablonu

7.1 Structure

7.1.1 Top Matter

Wypunktowanie

Jak to właściwie działa, gdy odwołuje się do obrazu ??

1 | Code goes here

Delete it after first real citation

Check after few pages if notes width command affects text

Remove code block after first real use

7.2 Minimalizacja parametrów funkcji plenoptycznej

Bibliografia

- [1] K. Academy. Dzielnik napięcia (z ang. voltage divider). <https://www.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider>, 2020.
- [2] Adafruit. Velostat - folia czuła na nacisk i zginanie. <https://www.adafruit.com/product/1361>, 2020.
- [3] 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>, 2020.
- [4] 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>, 2020.
- [5] Forbot. Rezystor - co warto wiedzieć. <https://forbot.pl/blog/leksykon/rezystor>, 2020.

TODOs:

■ TODO: ch:zagadnienia	3
■ Opis rękawic na rynku	4
■ INS - inertial navigation system	4
■ Jak dostępne rękawice omijają problem	4
■ Następny rozdział? - Dodaj sekcje mówiącym o wybraniu arduino ide oraz nie wybraniu istniejących sdk aby napisac własne polaczenie BLE w javie bez unity	5
■ Dodaj jakieś źródło flex sensor albo imu na palcach	9
■ TODO: ch:aplikacja	17
■ TODO: Krótki opis ó aplikacji"i rozdziale	17
■ TODO: Podsumowanie	18
■ Delete it after first real citation	19
■ Check after few pages if notes width command affects text	19
■ Remove code block after first real use	19