



SYSTEMY WBUDOWANE

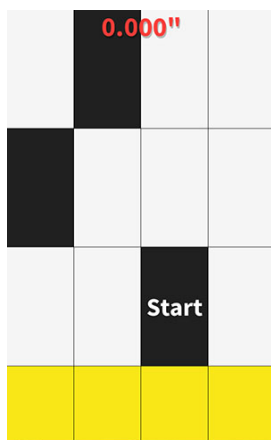
PianoBot

Projekt realizowany w ramach kursu Systemy Wbudowane w roku akademickim 2019/2020

Magdalena Wieczorek, Krzysztof Piaskowy

1 Wprowadzenie

Projekt polegał na zbudowaniu i napisaniu oprogramowania dla bota grającego w grę na urządzeniu z ekranem dotykowym (gra to aplikacja mobilna na systemie Android). Analizując obraz na wyświetlaczu, bot podejmuje decyzję o wykonaniu ruchu, poprzez emulowanie dotyku ekranu w odpowiednim czasie i miejscu na ekranie. Bot jest przystosowany do autonomicznego działania w aplikacji *Piano Tiles*.



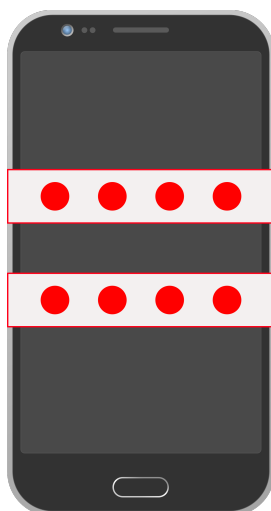
Rysunek 1: Aplikacja *TilePiano*

1.1 Zasada działania aplikacji

Bot operuje na urządzeniu mobilnym z systemem Android oraz zainstalowaną aplikacją *TilePiano*. Gra ta polega na przesuwaniu się z góry ekranu białych oraz czarnych płytek. Aby zdobyć punkt, należy w odpowiednim czasie nacisnąć na czarną płytkę. Wtedy wyzwala się nuta. Przy prawidłowym naciśnięciu sekwencji spadających płytek, usłyszymy popularny utwór czy piosenkę (*Tile - płytka*, *Piano - pianino*). Naciśnięcie na białą płytkę jest równoważne z błędem i przedwczesnym zakończeniem gry. Sukces następuje, gdy cała sekwencja nut zostanie wyzwolona w odpowiedniej kolejności, co jest równoważne z zakończeniem utworu.

1.2 Zasada działania systemu detekcji

Głównym kontrolerem systemu jest Arduino, gdzie znajduje się oprogramowanie do analizowania danych z sensorów i sterowanie serwo mechanizmami. Na ekranie są usytuowane dwa pasy złożone z 4 fotorezystorów ułożonych równolegle względem siebie jak na rysunku nr 2.

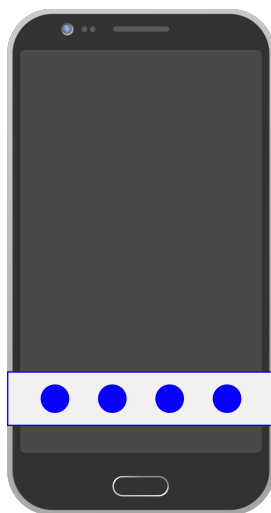


Rysunek 2: Rozmieszczenie fotorezystorów na wyświetlaczu

Dzięki zastosowaniu dwóch belek z fotorezystorami możliwe jest wyznaczenie prędkości poruszania się elementów na ekranie i dostosowanie odpowiedniego zachowania do zmieniających się warunków. Detekcja obiektu na ekranie odbywa się na podstawie danych o zmianach napięcia na fotorezystorach, ponieważ kolory na wyświetlaczu charakteryzują się różnym natężeniem światła emitowanego na danym obszarze w konkretnym kolorze. Z kolei zmiana natężenia światła padającego na fotorezystor wpływa na jego oporność, czyli w konsekwencji na odkładające się na nim napięcie. Pobierając wartości tego napięcia poprzez wejście analogowe na płytce Arduino można dokonać detekcji koloru obszaru znajdującego się pod fotorezystorem. Mając informację o stanie elementów wyświetlanych na ekranie możliwe jest podjęcie decyzji o wykonaniu ruchu.

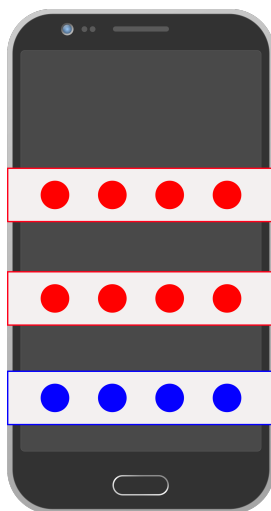
Za dotykanie ekranu są odpowiedzialne cztery przygotowane przez nas serwomechanizmy (*TowerPro SG-90*), na których znajduje się gąbka elektrostatyczna mogąca symulować dotyk, w miejscach reprezentowanych przez niebieskie kropki jak na rysunku nr 3.

Symulacja dotyku w jednym z 4 miejsc na ekranie będzie odbywać się przy pomocy przygotowanych przez nas fizycznych emulatorów dotyku, umieszczonych na ekranie w miejscach prezentowanych przez niebieskie kropki (rys.3):



Rysunek 3: Rozmieszczenie emulatorów na wyświetlaczu

Zasada działania emulatora jest następująca: Obecnie najczęściej wykorzystywane ekrany dotykowe są ekranami pojemnościowymi, co oznacza, że pod szklaną powłoką wyświetlacza znajduje się drobna siatka wykonana z przewodnika naładowana ładunkiem elektrycznym. Dotknięcie palcem wybranego obszaru powoduje przepłynięcie ładunku z siatki na ekranie do palca osoby dotykającej ekran. W tym momencie urządzenie jest w stanie zlokalizować miejsce, w którym nastąpił odpływ ładunków z ekranu i pobrać jego współrzędne. Wykorzystując ten efekt skonstruowaliśmy mechaniczny emulator dotyku, czyli serwo mechanizmy które poruszają ramieniem na którego końcu znajduje się uziemiona gąbka elektrostatyczna, która po zetknięciu z ekranem emuluje dotyk w wyznaczonym miejscu. W ten sposób uzyskujemy efekt analogiczny do dotknięcia ekranu palcem. Cały zestaw sensorów i emulatorów umieszczony na ekranie telefonu prezentuje się w następujący sposób:



Rysunek 4: Rozmieszczenie elementów detekcyjnych i emulujących na wyświetlaczu

System detekcyjny przed rozpoczęciem działania przechodzi kalibrację, podczas której uczy się, jak powiązać natężenie światła z konkretnym kolorem w zależności od oświetlenia znajdującego się w otaczającym go środowisku i ekranu, na którym

ma pracować. W tym celu stworzyliśmy aplikację wyświetlającą odpowiednią sekwencję kolorów dla każdego detektora. Podczas kalibracji układ wiąże odpowiednie poziomy natężenia z odpowiadającymi im kolorami.

2 Etapy realizacji projektu

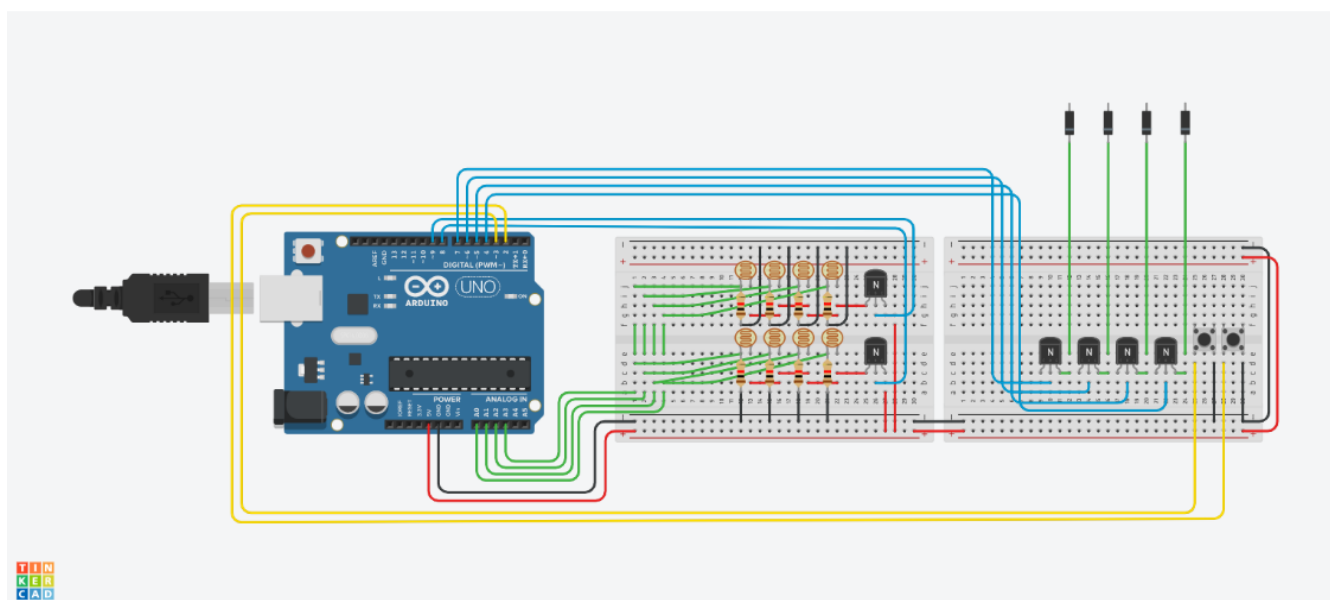
Plan realizacji projektu wyglądał następująco:

1. Zaprojektowanie schematu układu.
2. Połączenie ze sobą płytki Arduino oraz elementów elektronicznych.
3. Kalibracja fotorezystorów.
4. Oprogramowanie trybu ze stałą prędkością płytek.
5. Oprogramowanie trybu ze zwiększającą się prędkością płytek.
6. Stworzenie dokumentacji projektu (trwa przez cały okres tworzenia *PianoBota*).

3 Hardware / użyty sprzęt

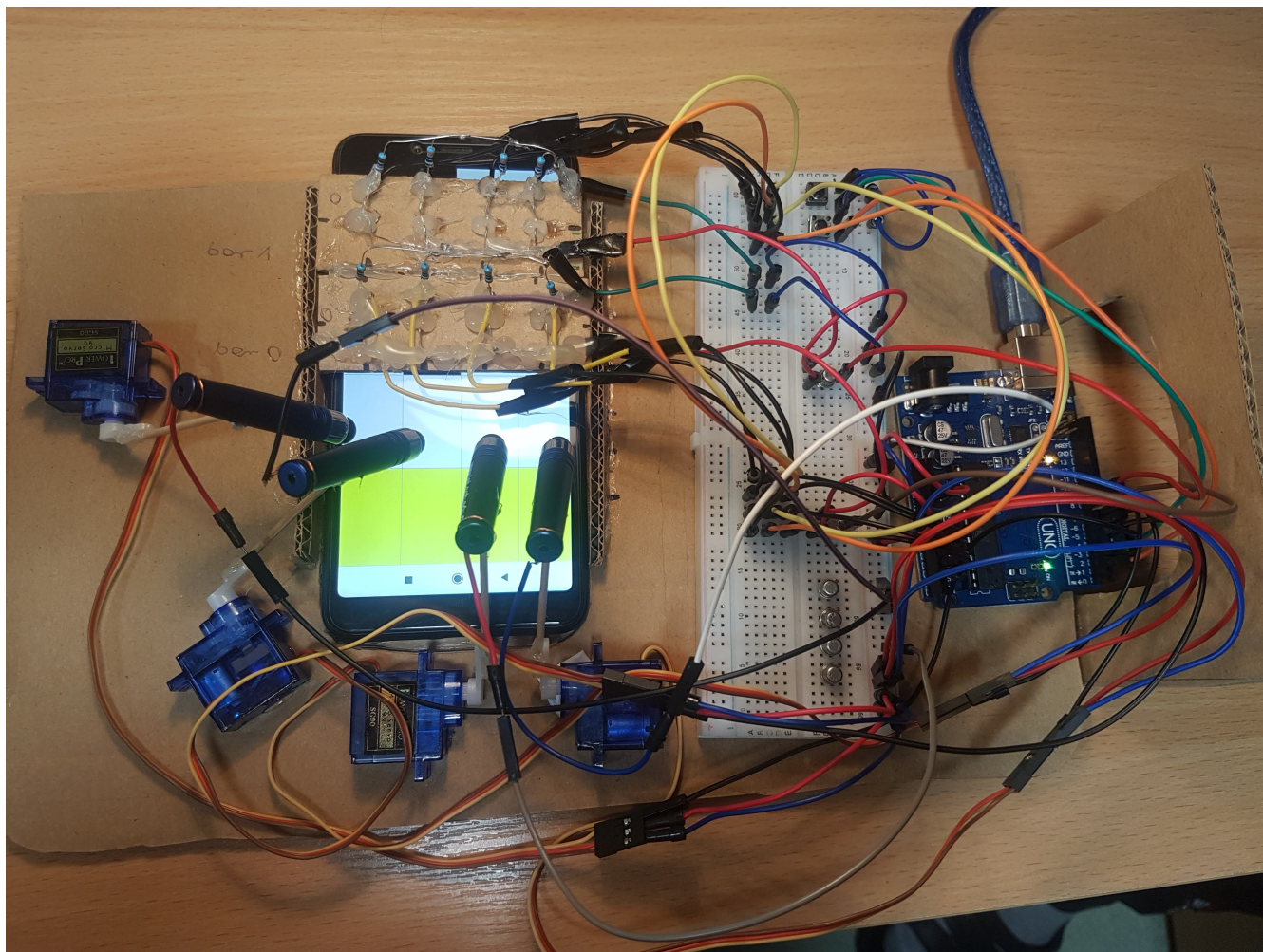
- Arduino UNO ATmega328
- Fotorezystory x8
- Tranzystory PN2222A x6
- Buttony x2
- Przewody miedziane
- Płytki stykowa
- Serwo TowerPro SG-90 x4

4 Schemat układu



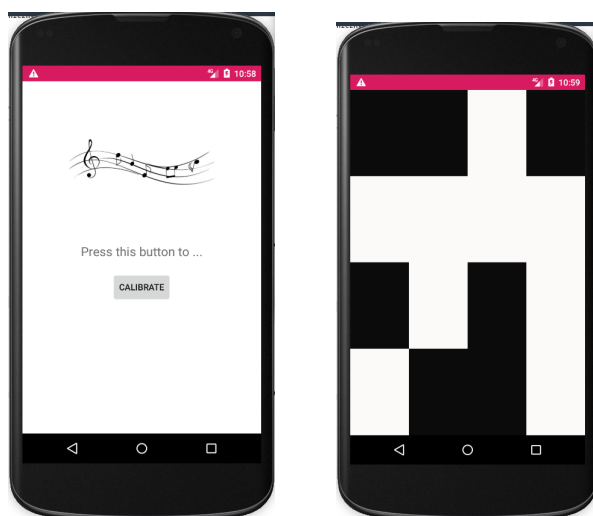
Rysunek 5: Schemat układu

5 Fizyczny układ



Rysunek 6: Zdjęcie układu po zbudowaniu

6 Aplikacja do kalibracji fotorezystorów



Rysunek 7: Aplikacja do kalibracji fotorezystorów

6.1 Zasada działania

Każdy z fotorezystorów w początkowej fazie projektu zostaje poddany kalibracji. Do tego celu napisaliśmy prostą aplikację symulującą grę *TilePiano* z taką modyfikacją, że płytka znajdująca się pod fotorezystorem jest cały czas czarna, a zmienia się tylko jej otoczenie w różnych konfiguracjach (określony górną i dolną granicą procent płytek jest czarny). Poniżej umieszczamy tylko najistotniejsze fragmenty aplikacji.

7 Instrukcja użytkowa

W celu uruchomienia bota należy zaopatrzyć się w urządzenie mobilne z systemem Android oraz zainstalowaną aplikację *Piano Tiles*. Cały projekt opracowywany był z wykorzystaniem aplikacji dostępnej w sklepie *Google Play* pod nazwą *Tap Black - Black Piano Tile*. Umieszczamy telefon pod dwoma rzędami fotorezystorów oraz gąbkami elektrostatycznymi przytwierdzonymi do serwomechanizmów (rys.4).

Włączamy Arduino z wgranym programem. Na naszym urządzeniu mobilnym uruchamiamy tryb klasyczny (tryb *Classic*). Aby uniezależnić się od obecności komputera można zaopatrzyć się w Powerbank, który będzie zasiliał układ.

8 Napotkane problemy i rozwiązania

Podczas realizacji projektu natknęliśmy się na trudności i przeszkody które zmusiły nas do zmienienia założeń i opracowania nowego rozwiązania które zastąpi poprzednie.

Pierwszą przeszkodą którą napotkaliśmy okazał się brak możliwości emulowania dotyku na ekranie smartfona poprzez uziemianie metalowych pastylek statycznie usytuowanych na ekranie (rys 8).

Zasada działania emulatora miała być następująca: Skonstruowaliśmy mechaniczny emulator dotyku składający się z 4 metalowych pastylek umieszczonych na ekranie telefonu. Każda pastylka miała być połączona z uziemieniem układu za pośrednictwem tranzystora sterowanego przez Arduino. W ten sposób uziemiając jedną z pastylek uzyskiwalibyśmy efekt analogiczny do dotknięcia ekranu palcem.

Okazało się jednak, że obecne ekrany dotykowe wyposażone są w bardziej inteligentne sterowniki niż ich kilkuletni poprzednicy. Gdyż aby ekran zarejestrował dotyk musi wykryć odpowiednio powiększający i zmniejszający się obszar z którego uchodzą ładunki, czyli dokładnie jak ma to miejsce podczas zetknięcia miękkiego opuszka palca z twardą powierzchnią ekranu. Początkowo ładunki odpływają z obszaru punktowego po czym w dalszej fazie styku palca i ekranu obszar ten się powiększa. Następnie podnosimy palec a obszar stopniowo się zmniejsza i takowego właśnie zjawiska oczekiwali sterowniki ekranu. Więc porzuciliśmy koncepcję z uziemianiem statycznych pastylek, a wykorzystaliśmy 4 serwa z ruchomymi ramionami na których końcu znajduje się uziemiony rysik z gąbką elektrostatyczna. W ten sposób można emulować dotyk który pod względem fizycznym działa dokładnie tak samo jak dotyk ludzkiego palca o ekran smartfona.

Kolejną przeszkodą jaką napotkaliśmy była sprawność emulatora dotyku, a mianowicie to, że działał mocno niedeterministycznie. Z nie do końca jasnych powodów dochodzi do tego, że mimo iż rysik ma kontakt z ekranem to nie wykrywa go jako dotyk. Był to dość poważny problem, jak poinformować kontroler o tym że nie powiodła się próba emulacji dotyku oraz o tym, że wymagane jest powtórzenie procesu. W tym celu wykorzystaliśmy drugą belkę fotorezystorów, i zapamiętujemy stany czujników z obu belek. I jeśli po próbie dotyku nie nastąpiło przesunięcie stanu z belki górnej na dolną to uznajemy, że próba się nie powiodła i ponawiamy proces.

W trakcie wytwarzania modelu natknęliśmy się również na dość prozaiczny problem a dokładniej nie wystarczającą ilość wejść analogowych w posiadanym modelu Arduino w stosunku do liczby posiadanych czujników analogowych - fotorezystorów. Doszliśmy do wniosku, że wykorzystamy dwa dodatkowe tranzystory do sterowania zasilaniem dla dwóch grup po 4 foto rezystory. Tranzystory podłączone do zasilania i wejścia cyfrowego. W ten sposób w momencie pomiaru doprowadzane jest napięcie do pierwszej grupy fotorezystorów, następnie poprzez zmianę stanów logicznych na tranzystorach napięcie zamiast do pierwszej grupy doprowadzane jest do drugiej. W ten sposób możemy dokonać pomiaru z 8 fotorezystorów niemal w tym samym czasie.