

## [Z7] KARTA PROJEKTU

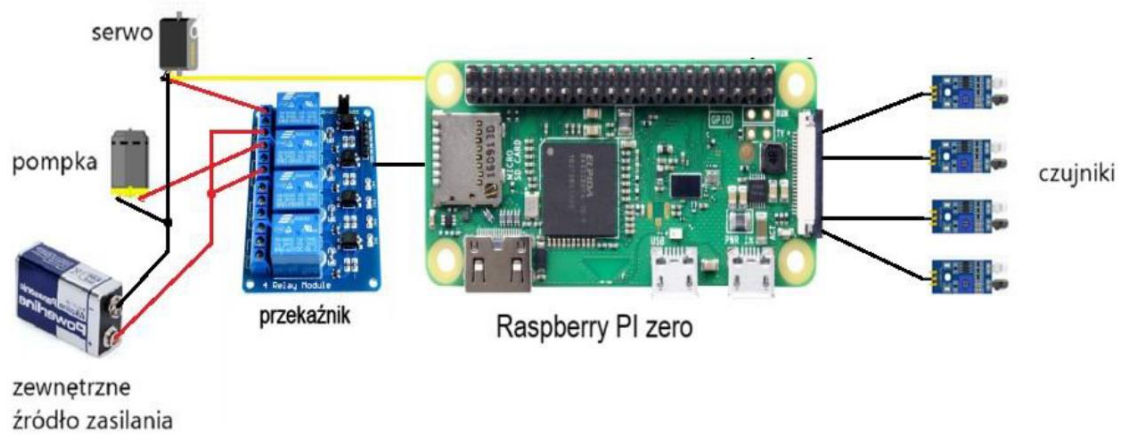
Grupa laboratoryjna: L6	Filip Rosiak 151799	
	Jakub Stefański 151876	
	Rafał Stróżyk 151793	
	Lena Woźniak 151904	
	Prowadzący zajęcia:	dr inż. Ariel Antonowicz
IMPREZOWY KAHOOT		
CEL PROJEKTU	Robot do rozlewania napojów do kieliszków. Robot przy pomocy czujników wykrywa obecność kieliszków. Jest połączony z aplikacją webową z interaktywną grą typu „kahoot”. Każdy z maksymalnie 4 graczy musi zaznaczyć poprawną odpowiedź na zadane pytanie - kto źle odpowie na pytanie temu robot za pomocą elementów wykonawczych nalewa tzw. „karniaka”.	
SCHEMAT POGLĄDOWY (UPROSZCZONY)		
<p>The diagram illustrates the hardware components and their connections. A Raspberry Pi Zero is the central microcontroller. It is connected to a servo motor, a pump, a relay module, and four distance sensors. A separate power source is connected to the pump and the relay module.</p>		
WYKORZYSTANA PLATFORMA SPRZĘTOWA, ELEMENTY POMIAROWE I WYKONAWCZE	- Raspberry PI zero - 4x czujniki odległości - serwomechanizm - pompka perystaltyczna do napoju - 2x przekaźnik	

## **1. Cel i zakres projektu.**

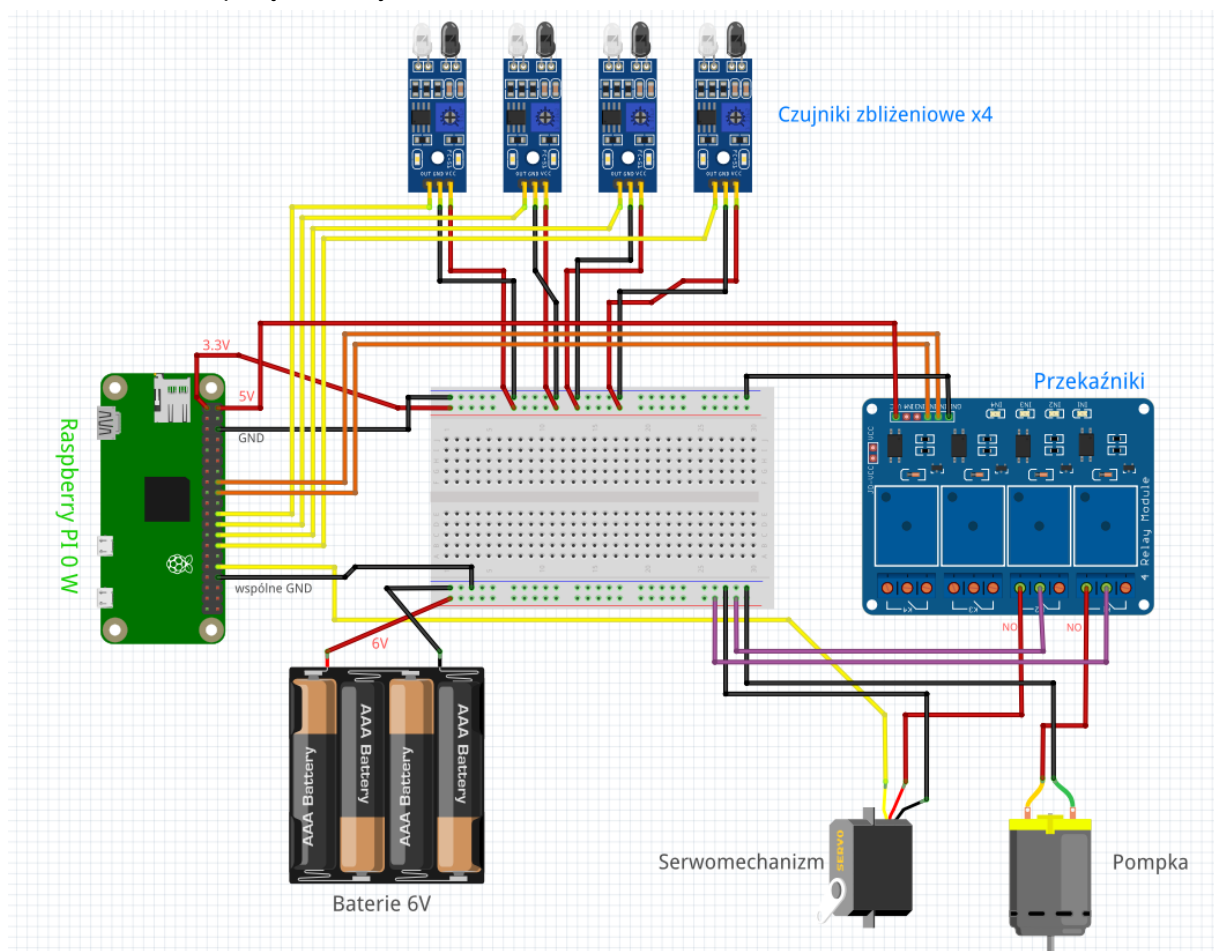
Projekt ma na celu stworzenie interaktywnego systemu, składającego się z elementów służących do rozlewania napojów do kieliszków oraz aplikacji webowej umożliwiającej grę w stylu „Kahoot”. Robot przy pomocy czujników zbliżeniowych wykrywać będzie obecność kieliszków, a gracze biorący udział w grze muszą udzielać poprawnych odpowiedzi na pytania, aby uniknąć „karniaków”, rozlewanych przy pomocy serwomechanizmu oraz pompki perystaltycznej. Wszystkie czujniki oraz elementy wykonawcze podłączone będą do Raspberry Pi Zero W. Aplikacja webowa „Kahoot” obejmuje interfejs użytkownika oraz komunikację z serwerem uruchomionym na RPi Zero W. Serwer obsługuje komunikację z klientami, logikę gry „Kahoot” oraz algorytm kontroli ruchu serwomechanizmu, uruchamiania pompki perystaltycznej oraz odczytywania wartości logicznej z czujników. Cały backend zostanie zrealizowany w języku python. Do części z grą „Kahoot” użyte zostaną biblioteki flask będąca frameworkiem do tworzenia aplikacji internetowych oraz flask\_socketio, która jest rozszerzeniem biblioteki flask dodająca m.in. obsługę komunikacji w czasie rzeczywistym przy pomocy WebSocketów. Część obsługująca czujniki oraz pozostałe elementy zostanie zrealizowana przy pomocy bibliotek RPi.GPIO oraz pigpio. Pierwsza z bibliotek użyta zostanie do odczytywania wartości logicznych z czujników oraz ustawiania wartości logicznych na pinach odpowiadających za uruchamianie zasilania serwa oraz pompki. Biblioteka pigpio zostanie natomiast użyta to generowania sygnału sterującego dla serwa. W przeciwieństwie do biblioteki RPi.GPIO, która może generować sygnał o wypełnieniu podanym jako %, pigpio dostarcza funkcję, która umożliwia generowanie sygnału o szerokości podanej w  $\mu s$ . Takie rozwiązanie zapobiega nieprawidłowemu generowaniu sygnału przy większym obciążeniu rdzenia procesora oraz eliminuje drgania serwa. Cały projekt integruje sprzęt i oprogramowanie, łącząc te elementy w celu stworzenia atrakcyjnej, imprezowej rozrywki.

## 2. Schemat.

Schemat ideowy:



Schemat połączeniowy:



### **3. Założenia projektowe a ich realizacja.**

Założeniem projektu było stworzenie interaktywnego systemu do rozlewania napoju to kieliszków, który byłby połączony z aplikacją webową oraz umożliwiał grę w stylu „Kahoot”. Robot miał przy pomocy czujników wykrywać kieliszki, a gracze mieli udzielać odpowiedzi na pytania przy pomocy aplikacji webowej. W przypadku złej odpowiedzi robot przy pomocy elementów wykonawczych miał polewać do kieliszków lub informować o jego braku. Serwer miał zostać napisany w języku Python, a dostęp do gry zrealizowany miał zostać poprzez stronę internetową zaprojektowaną specjalnie na telefony.

Z powodzeniem udało się zrealizować sprzętową stronę projektu, czujniki z powodzeniem wykrywają obecność kieliszków, a napisany algorytm porusza serwem i uruchamia pompkę. Aplikacja webowa również została pomyślnie zaimplementowana, obejmując przyjazny dla użytkownika interfejs oraz komunikację z serwerem. Z serwerem też wszystko się udało, z powodzeniem obsługuje sprzęt oraz logikę gry. Automatycznie przypisuje do graczy wolne id oraz odbiera od nich odpowiedzi na pytania, a następnie po rozlaniu karniaków wysyła kolejne pytanie, lub pokazuje finalny wynik rozgrywki. Udało się dodatkowo zaimplementować wyjątki takie jak rozłączenie gracza w trakcie rozgrywki, brak kieliszka, wyjęcie kieliszka w trakcie nalewania czy też próba dołączenia przez piątego gracza.

W przypadku dalszego rozwoju projekt ten ma ich naprawdę wiele. Można chociażby dodać opcje wyboru kategorii czy trudności pytań w celu zapewnienia satysfakcjonującego doświadczenia dla graczy lub różne tryby rozgrywki jak np. odpowiedź na czas, lub tryb drużynowy, gdzie drużyna pije gdy jedna z osób odpowie źle. Ponadto można by dodać coś w rodzaju edytora pytań, który pozwalałby użytkownikom tworzyć i zapisywać własne zestawy pytań, co jeszcze bardziej zwiększyłoby atrakcyjność gry. Kolejną opcją byłoby dodanie punktacji takiej jak w oryginalnej wersji gry gdzie szybsza odpowiedź nagradzana jest większą ilością punktów.

#### 4. Listing kodu.

tutaj umieść najważniejsze fragmenty kodu **wraz z komentarzami**.

```
@socketio.on('answer')
def handle_answer(data):
    player_id = data['player_id']
    answer = data['answer']

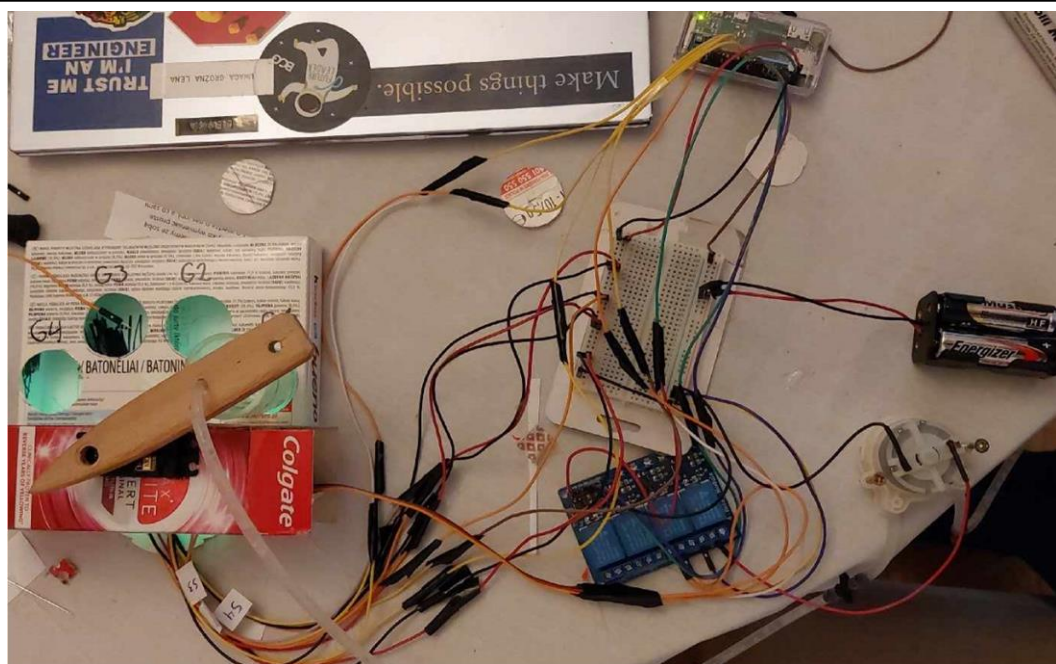
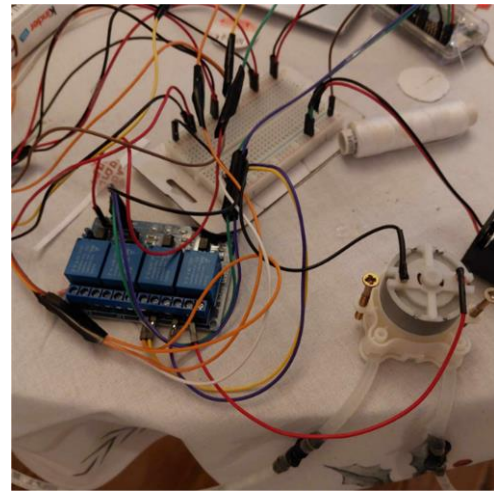
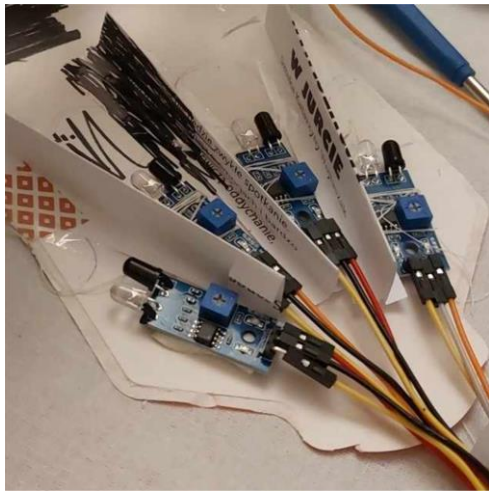
    # jeśli gracz jeszcze nie odpowiedział, dodanie go do listy których odpowiedzieli
    if player_id not in players_answered:
        players_answered.append(player_id)

    correct_answer = questions[current_question_index]['correct_answer']
    # Gdy gracz odpowiedział dobrze na pytanie dodawany mu jest punkt
    if answer == correct_answer:
        for player in players:
            if player['id'] == player_id:
                player['score'] += 1
                break
    else: # w przeciwnym wypadku dodawany jest do listy pijących
        players_answered_wrong.append(player_id)
    # wysłanie zaktualizowanej tabeli graczy
    emit('update_players', {'players': players}, broadcast=True)

    # gdy wszyscy gracze udzielili odpowiedzi
    if len(players_answered) == len(players):
        # gdy nieostatnie pytanie -> nadanie komunikatu, że wszyscy odpowiedzieli
        if current_question_index != question_limit - 1:
            emit('all_players_answered', broadcast=True)
        # rozlanie napojów po graczach z błędną odpowiedzią
        pourDrinks(players_answered_wrong)
        # wysłanie następnego pytania
        handle_next_question()
```

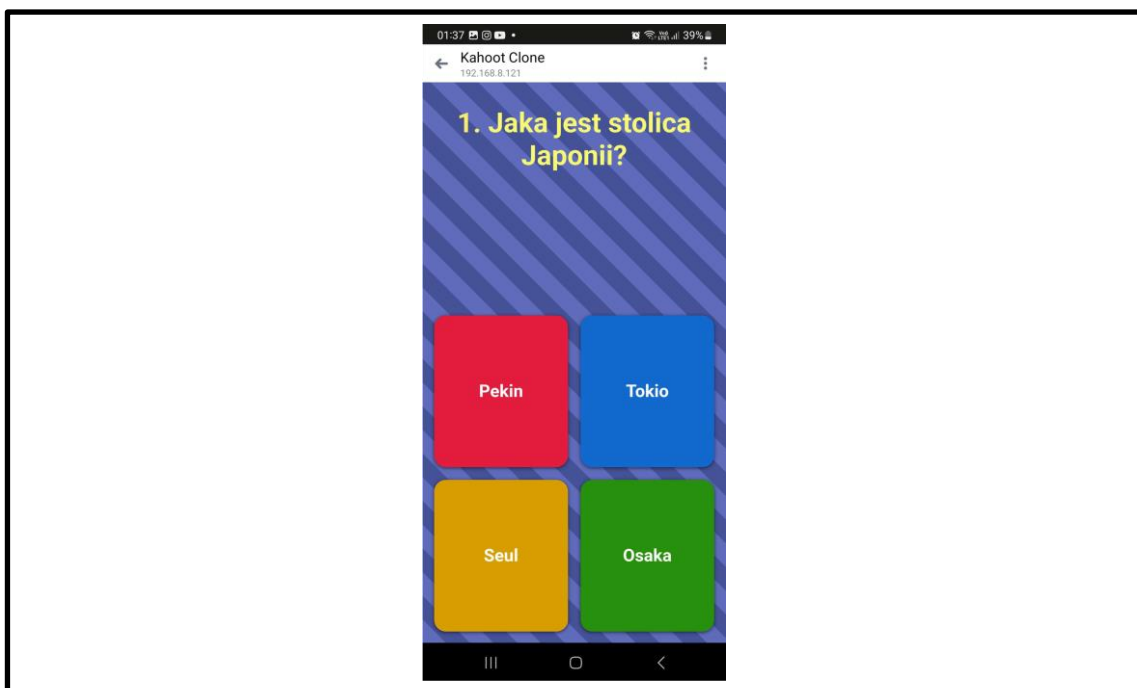
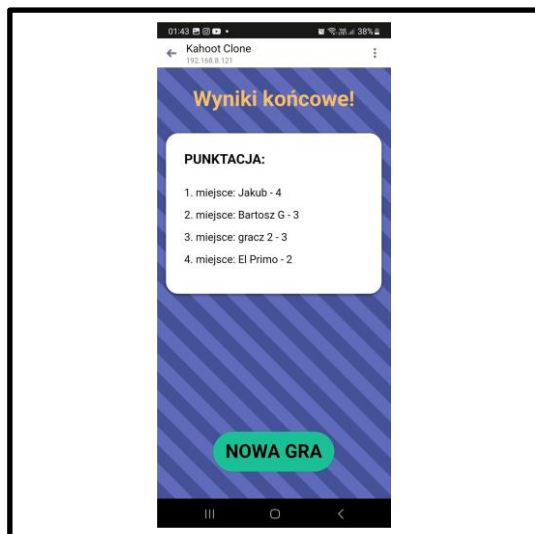
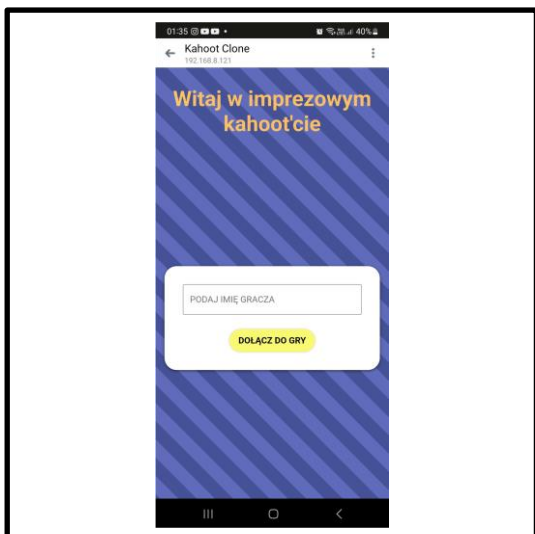
```
def pourDrinks(stupidPlayersIds):
    # dla każdego gracza z błędną odpowiedzią
    for playerID in stupidPlayersIds:
        # sprawdzenie czy kieliszek gracza jest na miejscu
        if GPIO.input(sensorPins[playerID - 1]) == GPIO.HIGH:
            emit('didnt_drink', {'id': playerID}, broadcast=True)
            # wstrzymanie nalewania do momentu ustawienia kieliszka
            while GPIO.input(sensorPins[playerID - 1]) == GPIO.HIGH:
                time.sleep(2)
        # obliczenie kąta ustawienia serwa
        slotAngle = 22.5 + 45 * (playerID - 1)
        # autorska funkca z kątem ustawienia serwa jako parametr
        servoGoToAngle(slotAngle)
        time.sleep(1)
        # nalewanie napoju
        pumpDrink(playerID)
        time.sleep(2)
    # po nalaniu wszystkim powrót serwa do pozycji zerowej
    if len(stupidPlayersIds) != 0:
        servoGoToAngle(0)
```

## 5. Zdjęcia zrealizowanego układu.





## 6. Zrzuty ekranu aplikacji (jeśli występują).



## **7. Podsumowanie i wnioski.**

Projekt „Imprezowy Kahoot” to innowacyjne połączenie robotyki oraz interaktywnej rozrywki. Naszym celem było stworzenie inteligentnego robota do rozlewania napoju, który współpracuje z aplikacją webową typu „Kahoot” tworząc ciekawą i pełną emocji grę dla użytkowników. Udało się spełnić wszystkie założenia, a nawet trochę więcej. Robot z sukcesem przy pomocy czujników wykrywa kieliszki lub informuje o ich braku oraz za pomocą pompki oraz serwa nalewa napój do kieliszków w przypadku, gdy gracz udzieli niepoprawnej odpowiedzi na zadane pytanie. Oczywiście, podczas realizacji tego projektu nie obeszło się bez komplikacji. Największy problem sprawiła nam integracja logiki gry z obsługą komponentów. Z osobna wszystkie działały jednak połączenie wszystkiego w jedno było większym wyzwaniem. W szczególności problem sprawiło nam serwo, które w połączeniu z biblioteką RPi.GPIO nie chciało działać gdy serwer był uruchomiony. Problemem okazał się sposób generowania sygnału przez bibliotekę. Jak się okazało biblioteka generuje ten sygnał poprzez swoją wewnętrzną funkcję, co w połączeniu z serwerem, który obciążał jedyny rdzeń sprawiło że serwo nie dostawało właściwego sygnału. Problem udało się rozwiązać przy pomocy innej biblioteki – pigpio, która generuje sygnał przy pomocy hardware’u, a nie software’u. Przy pozostałych etapach problemy występowały rzadko, a jeśli już wystąpiły ich naprawa nie zajmowała dużo czasu.

Projekt „Imprezowy Kahoot” był dla nas fascynującym wyzwaniem jak i cenną nauką związaną z rozwojem zaawansowanego systemu wbudowanego. W trakcie jego realizacji odczuwaliśmy satysfakcję, widząc jak nasz pomysł na inteligentnego robota do rozlewania napoju nabierał realnych kształtów i każdy etap budowy i programowania robota był dla nas ciekawym doświadczeniem. Ostatecznie uważamy, że projekt stanowi fascynujące połączenie robotyki, technologii interaktywnej oraz zabawy.