

# Karta projektu z wykładu

## Systemy mikroprocesorowe - 2021

Temat projektu: **Matryca LED sterowana smartfonem**

Imię i nazwisko: **Dominik Bogielczyk**

Nr albumu: **144435**

Kierunek: **AiR**, grupa: **A3**

### 1. Opis projektu

Celem projektu było zbudowanie układu sterowania czterema matrycami LED 8x8 połączonymi szeregowo za pomocą własnej aplikacji na Androida.

Funkcjonalności:

- wyświetlanie gif-ów: serce, pacman, schody, losowy wzór,
- gra snake – sterowanie przyciskami w aplikacji lub ruchem smartfonu,
- losowanie: kostki do gry, liczba 0-99 (kostki także poprzez potrząśnięcie smartfonem),
- wyświetlanie tekstu wprowadzonego w aplikacji.

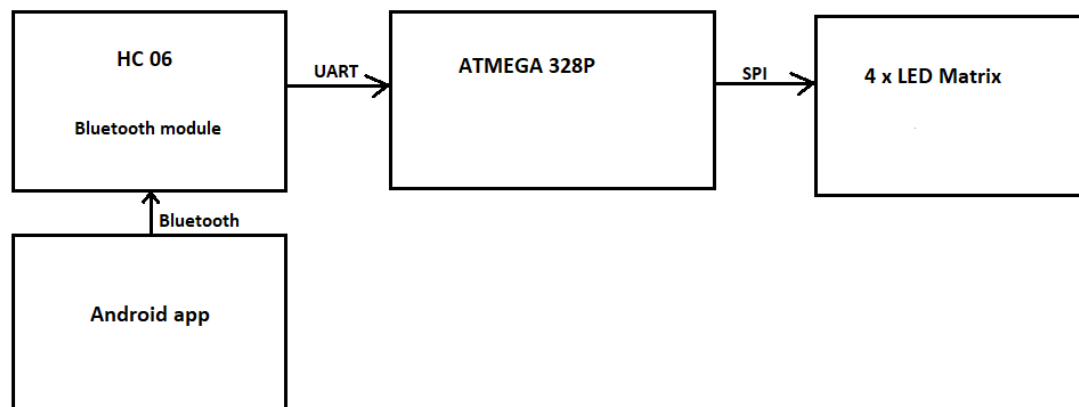
4 matryce połączone szeregowo tworzą 8x32 pixeli do sterowania.

Sterowanie w trybie gif-ów oraz funkcji losowania dla każdej z matryc niezależnie.

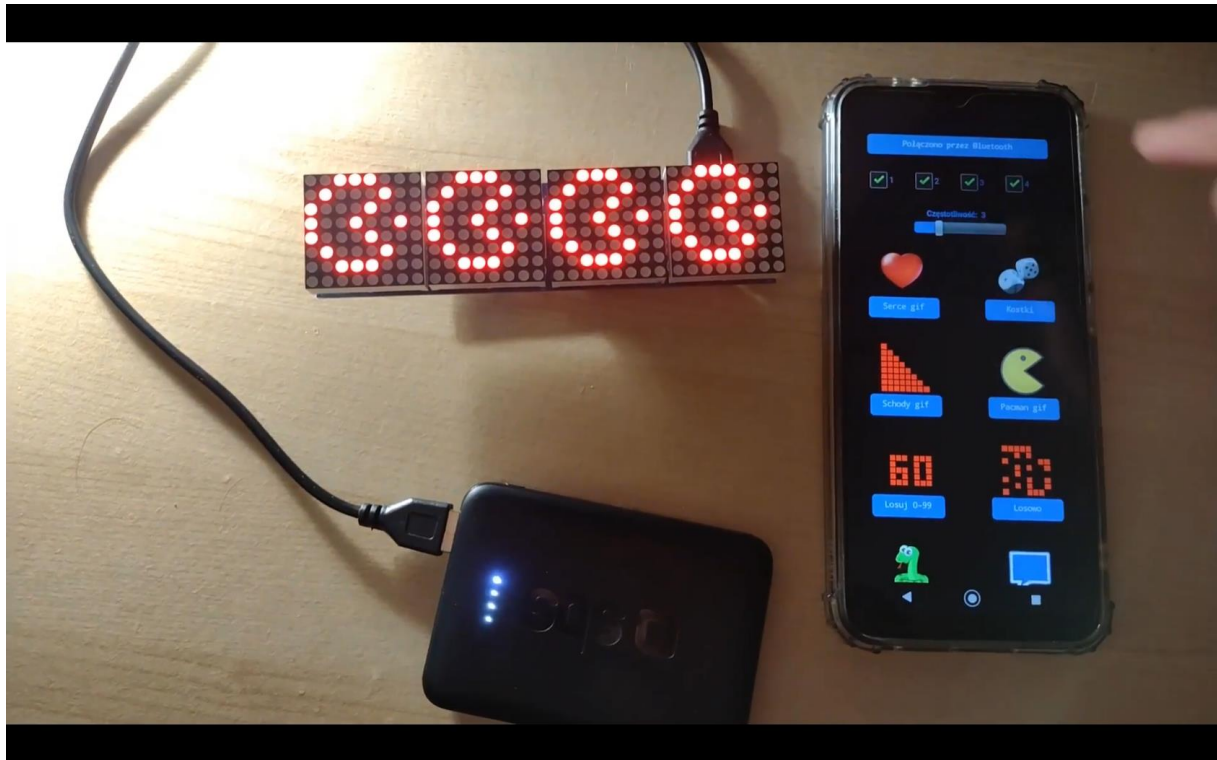
Repozytorium projektu na Githubie:

<https://github.com/DominikBogielczyk/LED-Matrix-Remote-Control>

Schemat blokowy systemu:



Zdjęcie układu:



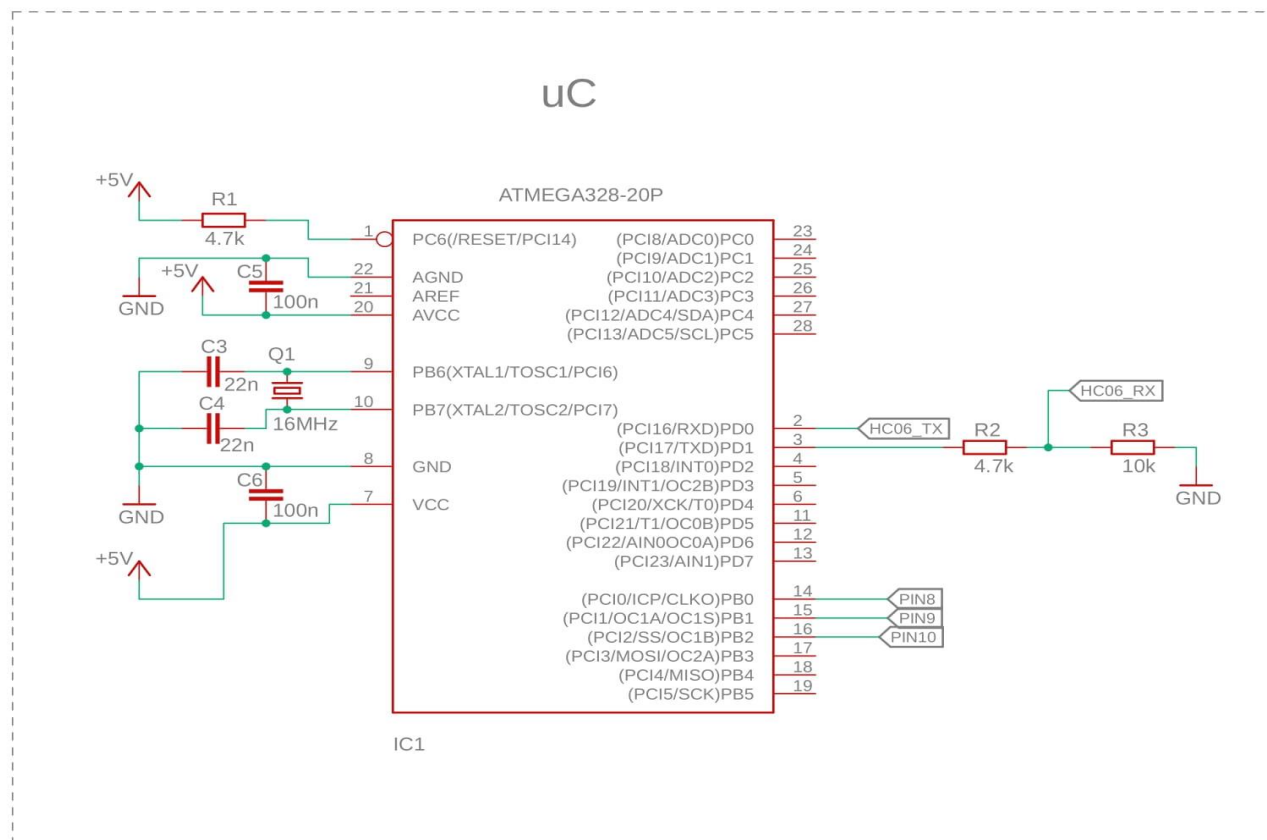
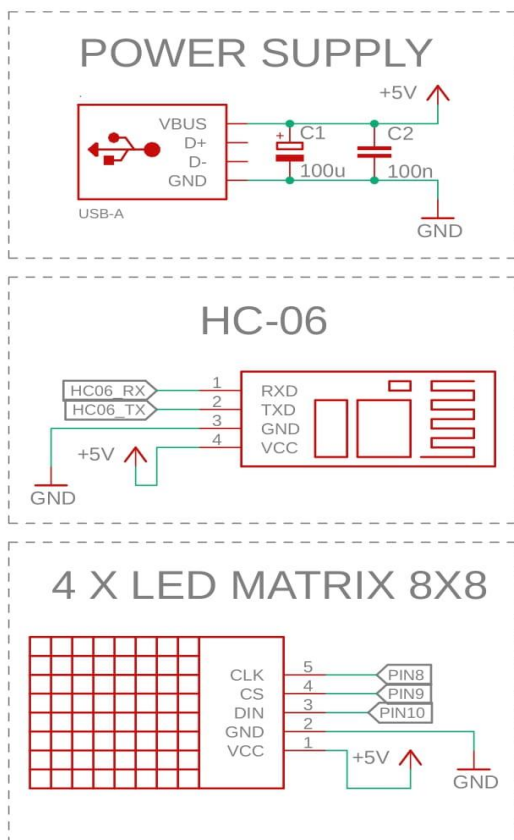
## 2. Hardware

Zastosowane elementy elektroniczne:

- mikrokontroler ATmega 328P (w wersji prototypowej na płytce Arduino Uno),
- moduł Bluetooth HC-06,
- 4 matryce LED 8x8 ze sterownikami MAX7219,
- kondensatory oraz rezystory,
- rezonator kwarcowy 16MHz,
- zasilanie wejściem USB-A.

Do obsługi układu wymagane jest urządzenie z systemem Android.

Schemat układu:



### 3. Software

Kod dotyczący działania mikrokontrolera omówię tylko ogólnie, gdyż jest on dostępny na Githubie, natomiast umieszczona aplikacja na Androida to tylko plik instalacyjny apk, więc tę część software omówię bardziej szczegółowo.

#### 3.1. Mikrokontroler

Zaprogramowany z wykorzystaniem Arduino IDE.

Biblioteka do obsługi matryc LED: LED Control.

Cały kod:

[Github 8x8 and HC 06.ino](#)

Najważniejsze funkcje:

- `receive_data()` – obsługa otrzymanych danych UARTem,
- `print_pattern(byte *value, uint8_t addr)` – ustawienie stanów pixeli matryc,
- obsługa wyświetlanych wzorów
  - `heart()` – gif serce,
  - `pacman()` – gif pacman,
  - `draw()` – losowanie liczby 0-99,
  - `dice()` – losowanie kostkami,
  - `stairs()` – gif schody,
  - `snake()` – gra Snake.

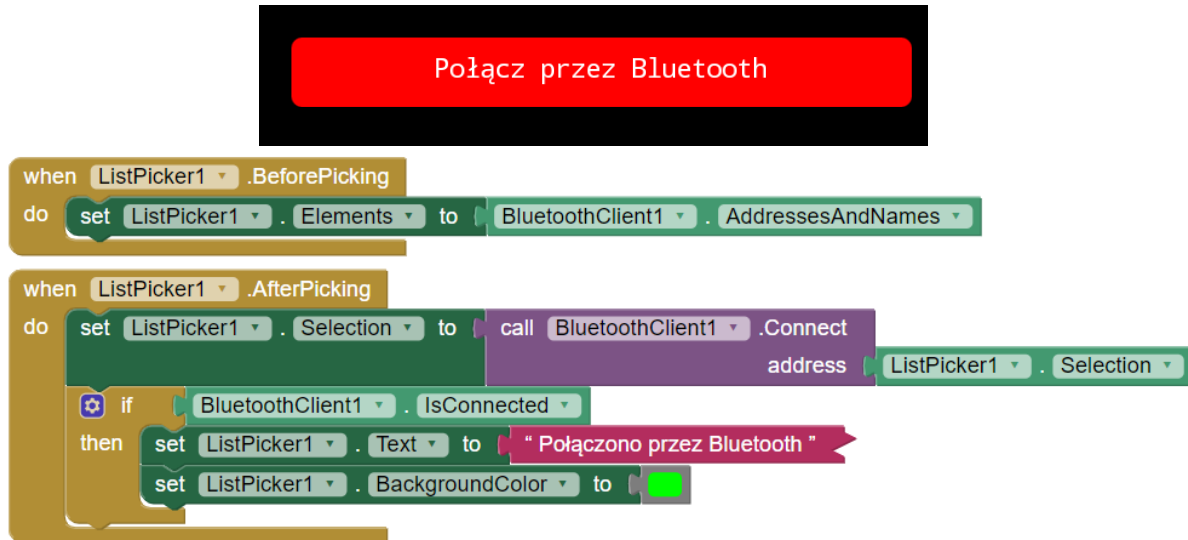
Funkcjonalności losowania wywoływane są jednorazowo po ich wybraniu, natomiast gify wykonywane są w pętli zgodnie z globalnie zadeklarowanymi licznikami numeru ramki. Informacja o aktualnym trybie przechowywana jest w zmiennej *modes*, dzięki czemu możemy sterować niezależnie każdą macierzą pixeli. Informacja o tym, której macierzy stan chcemy zmienić (poprzez zaznaczenie checkboxów **1, 2, 3, 4**) przechowywana jest w zmiennej tablicy boolowskiej *matrixes*.

### 3.2. Aplikacja na Androida

Stworzona w narzędziu przeglądarkowym MIT App Inventor.

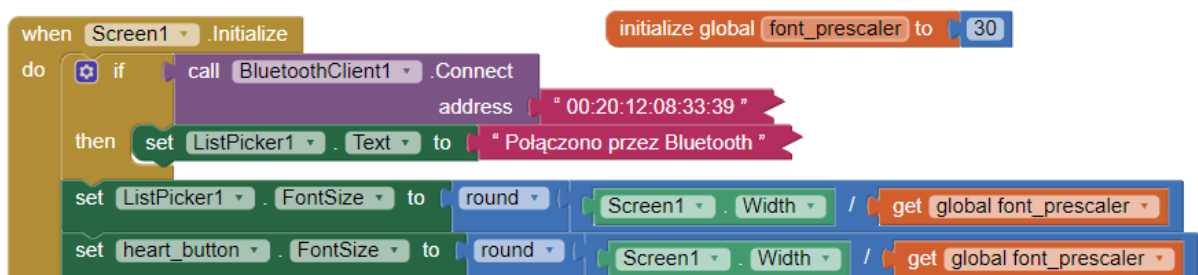
**Najistotniejsze bloki aplikacji:**

- połączenie przez Bluetooth

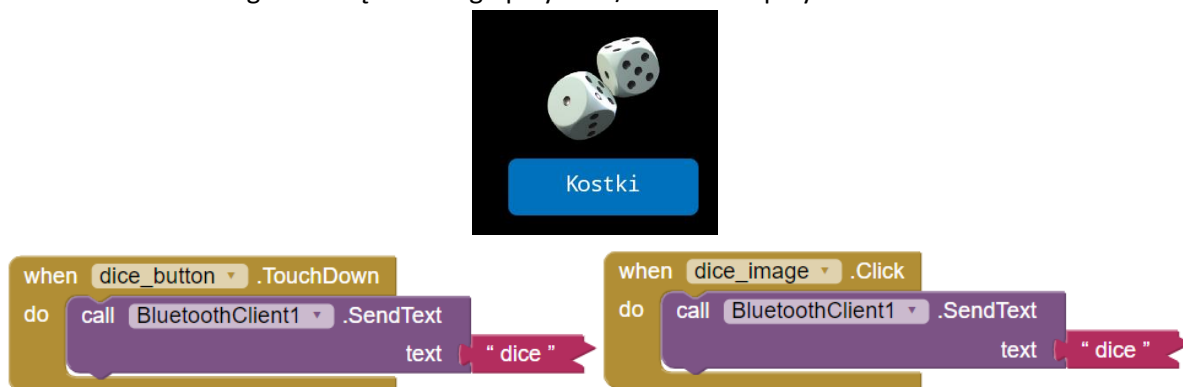


Powyższe łączenie wywołujemy kliknięciem **Połącz przez Bluetooth** oraz wybranie modułu HC-06 jako urządzenia, z którym chcemy się połączyć. Należy to zrobić jeśli połączenie nie nastąpiło automatycznie, a automatyczne łączenie wykonywane jest przy inicjalizacji ekranu z zadeklarowanym adresem modułu bluetooth HC-06.

Przy okazji na poniższym screenie widać również funkcję użytą do przeskalowania tekstu, tak aby aplikacja wyglądała podobnie dla urządzeń z różną rozdzielczością. Każdy napis jest tak przeskalowany.



- obsługa wciśnięcia danego przycisku/obrazka na przykładzie **Kostki**

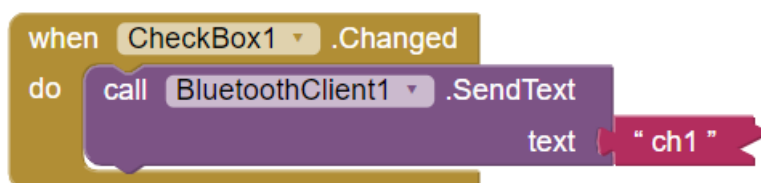


Bloczek .SendText wysyła ciąg znaków przez Bluetooth do modułu HC-06, a następnie te dane przesyłane są przez UART do mikrokontrolera.

Obsługa tego zdarzenia przez Atmegę:

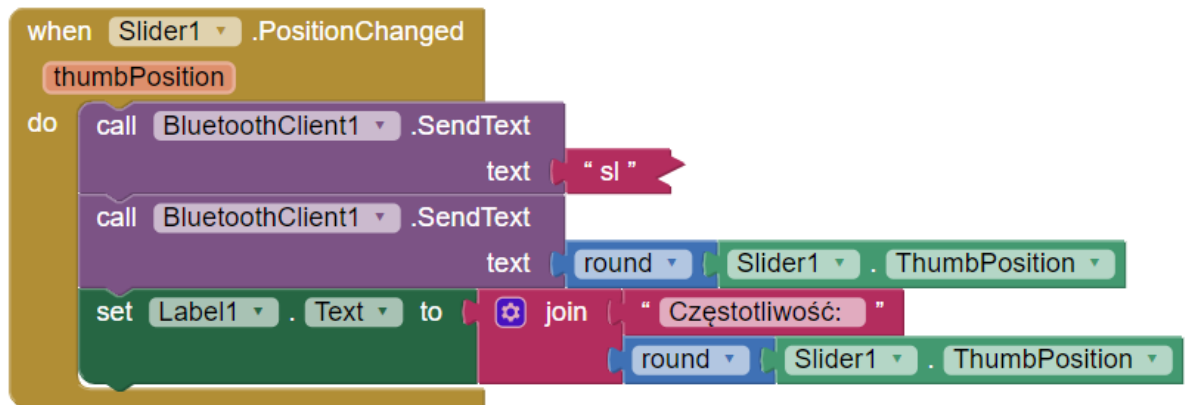
```
190     if (received_data.indexOf("dice") != -1)
191     {
192         dice();
193     }
```

- wybranie danej matrycy, na której dokonujemy zmian na przykładzie **1**



- slider – zmiana częstotliwości wyświetlanych gif-ów





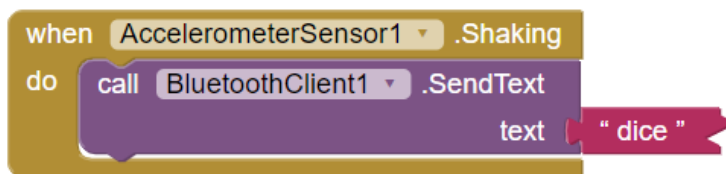
Obsługa otrzymanych danych przez mikrokontroler:

```

183  if (received_data.indexOf("sl") != -1) //SLIDER HANDLE
184  {
185      if (prescaler >= 1 and prescaler <= 9)
186      {
187          prescaler = received_data[received_data.indexOf("sl") + 2] - 48; //from ACII table '0' = 48, '1' = 49, ...
188      }
189  }

```

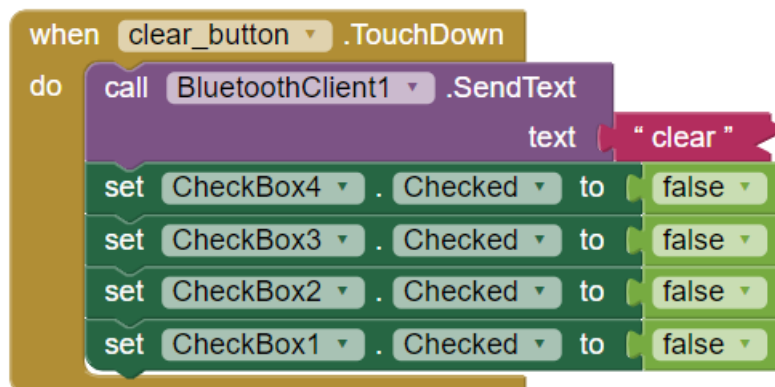
- obsługa potrząśnięcia urządzeniem w celu losowania kostek



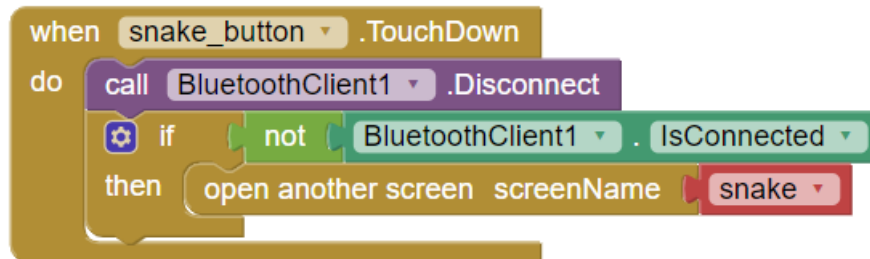
Środowisko MIT App Inventor ma wbudowaną funkcję obsługującą akcelerometr urządzenia, dzięki czemu wystarczy jeden bloczek do utworzenia funkcji wywoływanej przy potrząśnięciu urządzenia.

**Uwaga.** Niektóre urządzenia mogą mieć domyślnie ustawione jakieś akcje przy potrząśnięciu nimi np. smartfon może włączać wtedy latarkę.

- przycisk **Wyczyść**

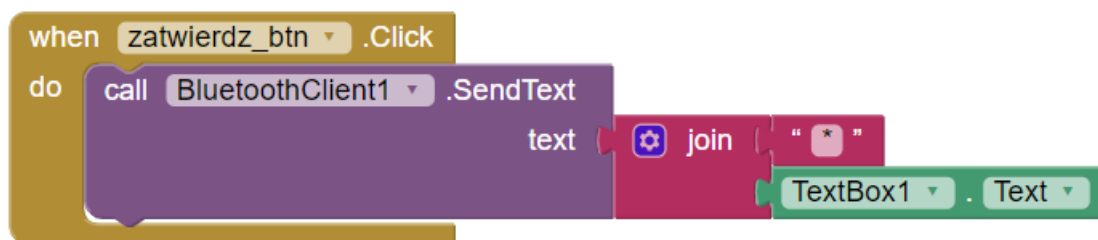


- otworzenie innego ekranu na przykładzie **Snake**



W tym środowisku każdy ekran jest niczym nowa aplikacja, dlatego wymagane jest łączenie przez Bluetooth od nowa lub wyświetlanie wszystkiego na jednym ekranie i obsługa grafik oraz przycisków przez odpowiednie zmienianie ich widoczności. Ja zdecydowałem się na tworzenie nowych ekranów i nowe połączenie Bluetooth po każdej zmianie ekranu. Aplikacja składa się z 3 ekranów: ekranu głównego, ekranu do gry snake oraz ekranu do wpisywania tekstu.

- przycisk **Zatwierdź** – ekran wpisywania tekstu



Znak \* przesyłany, aby wiedzieć jak interpretować otrzymany przez Atmegę ciąg znaków.

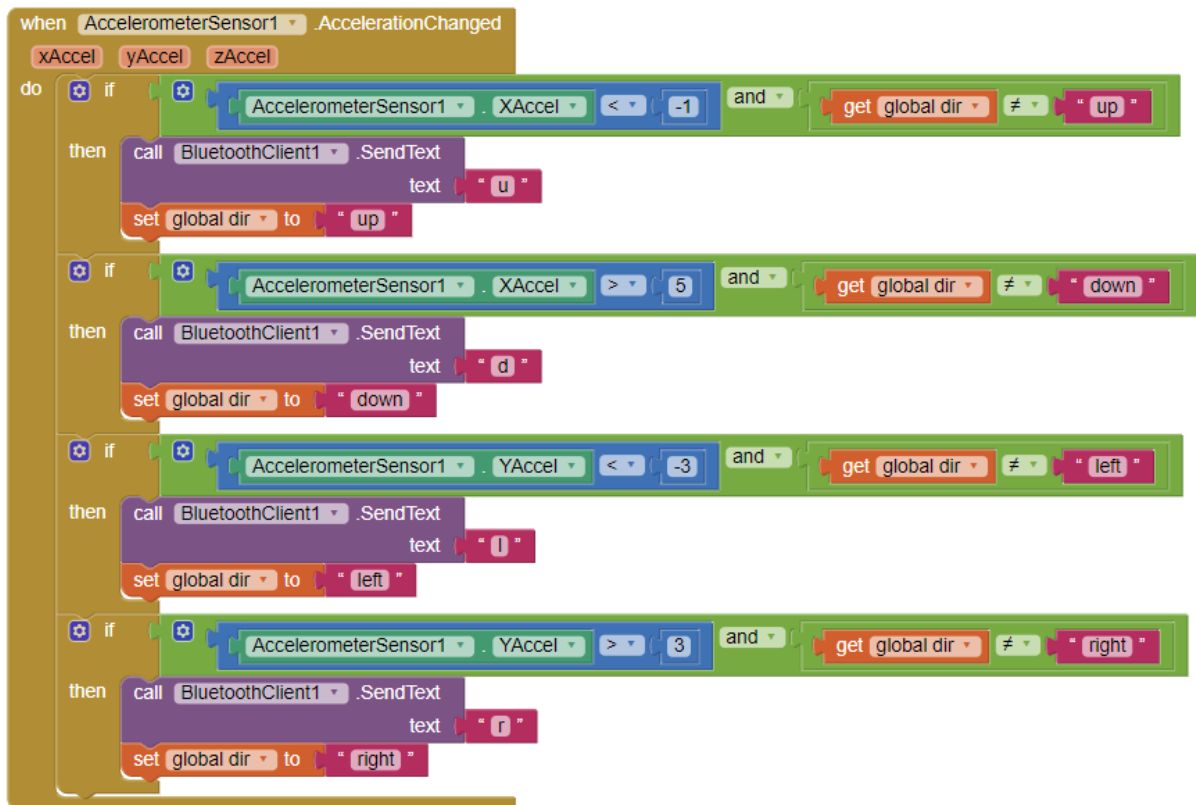
Obsługa tego zdarzenia przez mikrokontroler:

```
202     else if (received_data.indexOf("*") != -1)
203     {
204         entered_text = received_data.substring(received_data.indexOf("*") + 1);
205         entered_text.toUpperCase();
206         text_i = 0;
207     }
```



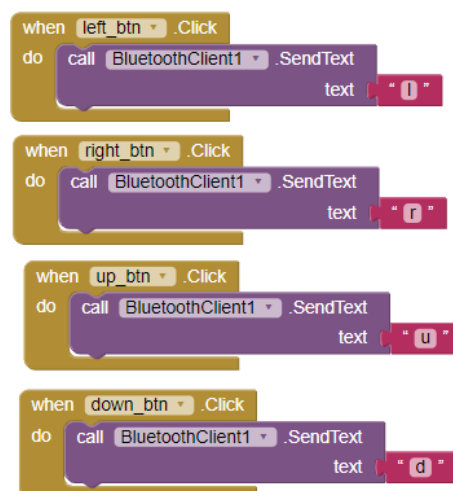
- sterowanie w grze snake

- poprzez pochylanie smartfonu



AccelerometerSensor1 to wbudowany w środowisko MIT App Inventor komponent umożliwiający wykorzystanie wbudowanego w urządzenie z systemem Android akcelerometru.

- sterowanie przyciskami



Obsługa otrzymanych danych przez mikrokontroler:

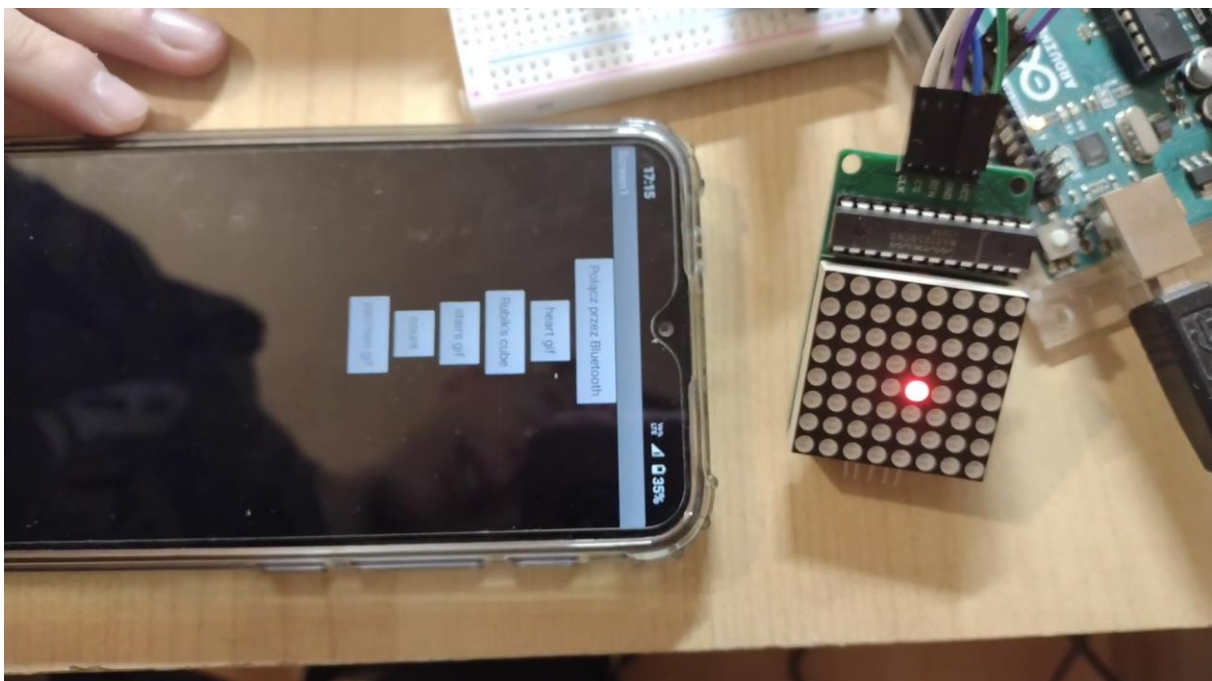
```
521     if ((digitalRead(right_switch) == 0 or dir_cmd == 114) and dir != "left")
522     {
523         dir = "right";
524     }
525     else if ((digitalRead(left_switch) == 0 or dir_cmd == 108) and dir != "right")
526     {
527         dir = "left";
528     }
529     else if ((digitalRead(up_switch) == 0 or dir_cmd == 117) and dir != "down")
530     {
531         dir = "up";
532     }
533     else if ((digitalRead(down_switch) == 0 or dir_cmd == 100) and dir != "up")
534     {
535         dir = "down";
536     }
```

Wartości 114, 108, 117, 100 wynikają z kodów ASCII znaków r, l, u, d.

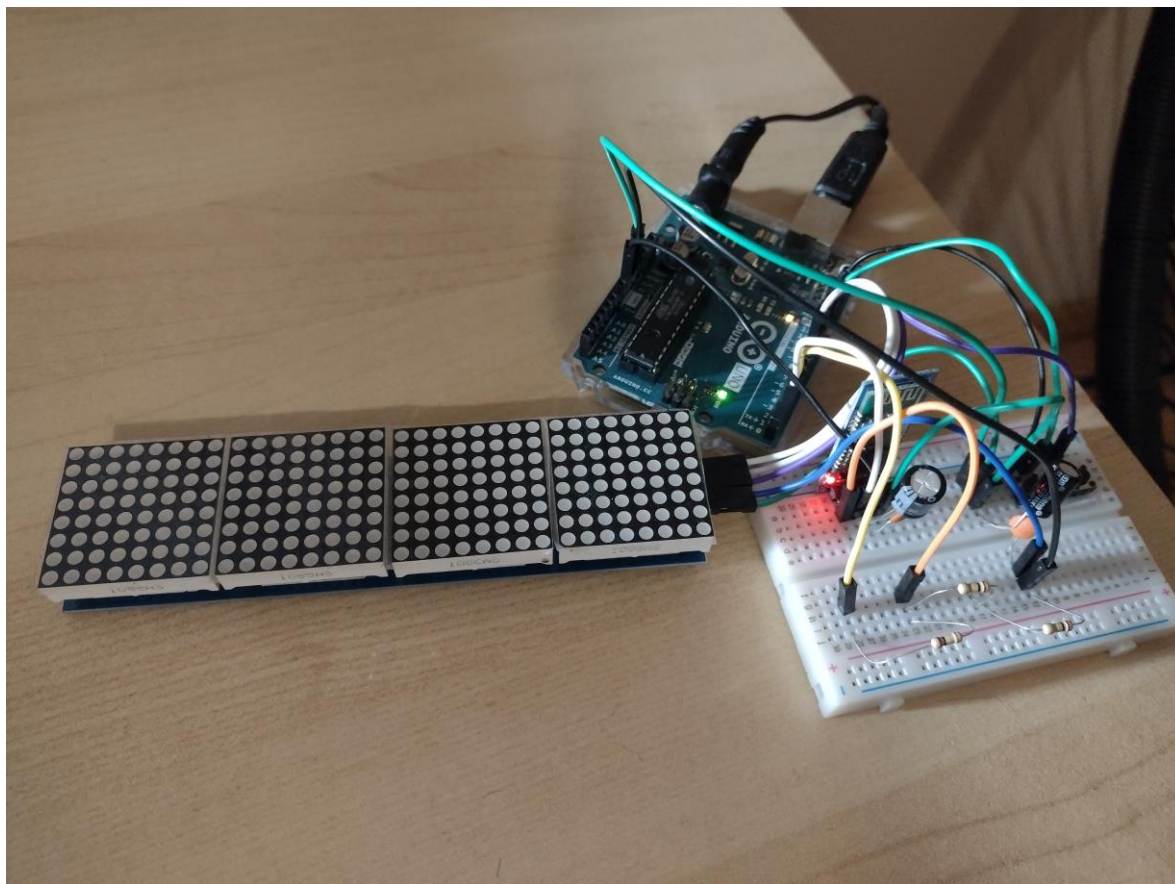
Mikrokontroler zaprogramowany jest także do sterowania za pomocą przycisków podłączonych do wejść cyfrowych, jednak zrezygnowałem z tego sposobu sterowania w ostatecznej wersji i układ nie posiada fizycznie tych przycisków.

#### 4. Proces montażu i programowania systemu

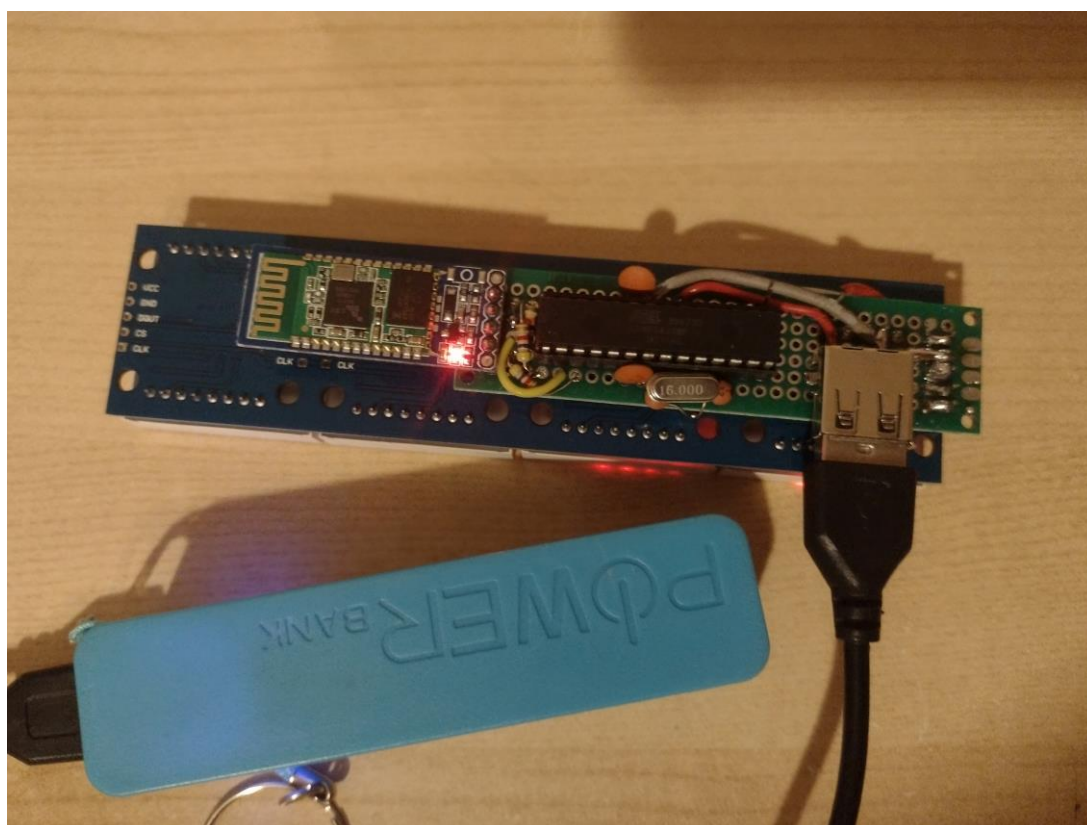
W pierwszym etapie układ został zbudowany na płytce stykowej z wykorzystaniem Arduino UNO oraz posiadał tylko 1 matrycę LED 8x8, a aplikacja zawierała jedynie kilka prostych przycisków:

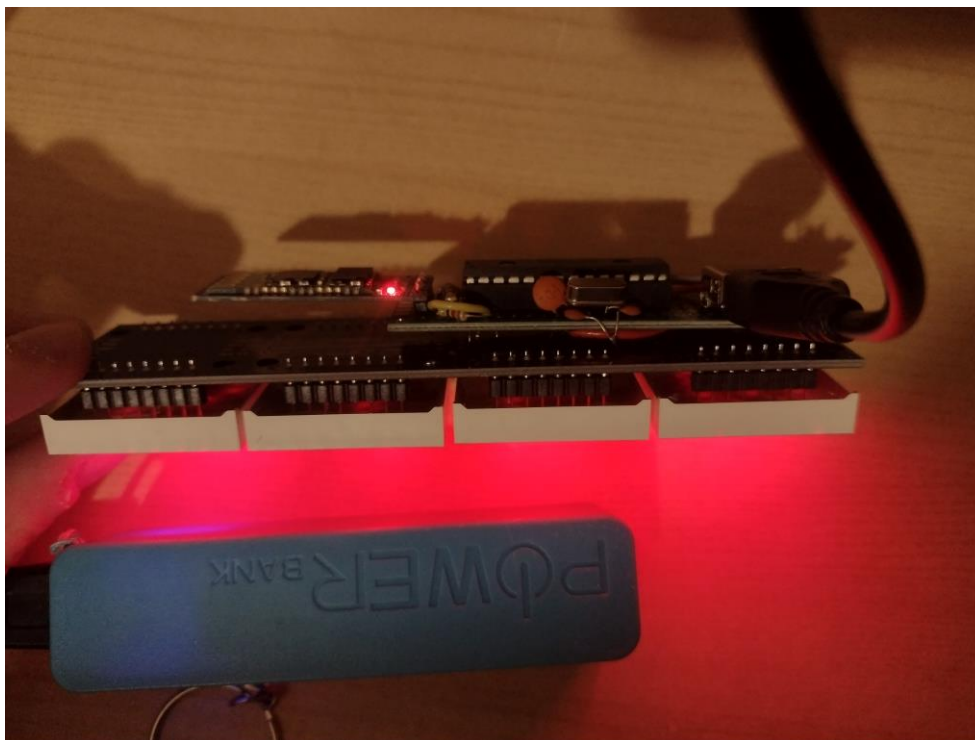


Następnie w celu wyświetlania tekstu oraz ogólnie większych możliwości dokupiłem 4 matryce LED połączone szeregowo, a więc tworzące w sumie 8x32 pixeli:



W ostatnim etapie układ został przebudowany, aby korzystać z samej Atmegi 328P i po sprawdzeniu poprawności działania zlutowany na płytce uniwersalnej:





## 5. Weryfikacja poprawności działania układu

Testy działania systemu można prześledzić oglądając filmik:

[Filmik w serwisie youtube](#)

## 6. Obsługa układu

1. Należy sparować nasze urządzenie z systemem Android z modulem Bluetooth dostępnym pod nazwą HC-06.
2. Należy połączyć się z modulem Bluetooth klikając czerwony przycisk **Połącz przez Bluetooth**. Połączenie może nastąpić automatycznie, wtedy nie musimy wykonywać tego ręcznie. Gdy połączenie przebiegło pomyślnie przycisk **Połącz przez Bluetooth** zmieni kolor na niebieski. Ten punkt należy powtarzać po każdej zmianie wyświetlanego ekranu (menu, snake, text).

Przycisk przed połączeniem:



Przycisk po połączeniu:



Błąd połączenia, gdy matryca LED jest nie zasilona, bądź nie włączono Bluetooth w urządzeniu sterującym:

Error 515: Not connected to a Bluetooth device.

Urządzenie, które należy wybrać:



Gdy nie sparujemy urządzeń wcześniej, to będzie ono niewidoczne w tym miejscu.

### 3. Ekran główny

- zmiana wzoru

Najpierw należy wybrać, na których matrycach chcemy dokonać zmian, a następnie wybrać funkcję wyświetlania poprzez kliknięcie na grafikę bądź przycisk.

wybranie matrycy, na którym chcemy dokonać zmian



- wyczyszczenie ekranu

Ekran możemy wyczyścić poprzez przycisk **Wyczyść**.

- uruchomienie innego ekranu: **Snake / Wpisz tekst**

Po wybraniu jednej z tych opcji otworzony zostanie nowy ekran, dlatego należy ponownie połączyć się przez Bluetooth.



#### 4. Snake

Jeżeli poprawnie połączyliśmy się przez Bluetooth to uruchomiona zostanie gra Snake. Sterować możemy poprzez strzałki lub poprzez pochylanie urządzenia sterującego, jeśli wyposażone jest ono w akcelerometr.

W przypadku przegranej, gra jest zatrzymana na 5 sekund, po czym następuje nowa gra. Wyjść z rozgrywki można w każdym momencie poprzez przycisk **Wróć do menu**.

#### 5. Wpisz tekst

Po pomyślnym połączeniu przez Bluetooth możemy ustawić tekst wyświetlany przez matryce. Aby tekst wgrał się na ekran należy go zatwierdzić.



Powrót do menu głównego nie powoduje wstrzymania wyświetlania tekstu, aby wstrzymać tę funkcjonalność należy wyczyścić ekran przyciskiem **Wyczyść**.

## 7. Literatura

1. Horowitz, W. Hill: Sztuka Elektroniki, tom 1 i 2, WKiŁ, Warszawa, 1995
2. Simon Monk: Arduino dla początkujących. Kolejny krok, 2015

Źródła internetowe:

- forbot.pl
- elektroda.pl
- appinventor.mit.edu
- docs.arduino.cc