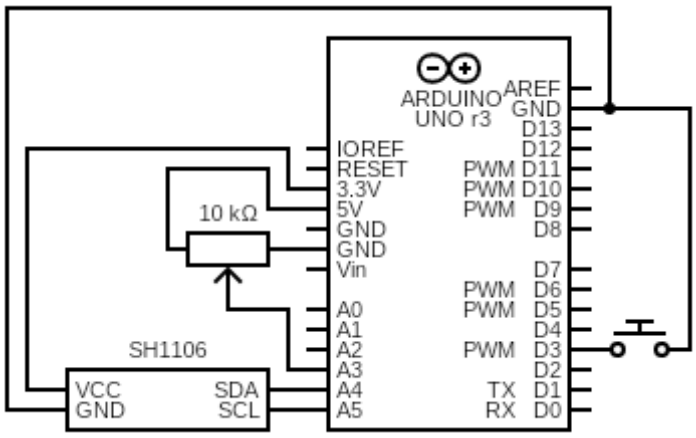
Eksperyment 13: Kółko i krzyżyk

Współautor i pomysłodawca: Jan Bensz

Ten program jest implementacją gry w kółko i krzyżyk na mikrokontrolerze z wykorzystaniem wyświetlacza OLED i przycisku do interakcji. Oto krótki opis poszczególnych części programu:

1. **Biblioteki i deklaracje**
   - Wykorzystuje bibliotekę "lcdgfx.h" do obsługi wyświetlacza OLED.
   - Inicjalizuje obiekt DisplaySH1106_128x64_I2C i NanoCanvas do rysowania na ekranie.
   - Deklaruje zmienną plansza reprezentującą planszę gry, oraz inne zmienne i funkcje pomocnicze.
2. **Funkcje pomocnicze**
   - odNowa() - Czyści planszę gry i ustawia zmienną czyNowaGra na true.
   - czyWygrana() - Sprawdza warunki wygranej w grze w kółko i krzyżyk.
   - czyRemis() - Sprawdza, czy gra zakończyła się remisem.
3. **Setup()**
   - Konfiguruje pin 3 jako wejście z wewnętrznym pull-up.
   - Inicjalizuje wyświetlacz OLED.
   - Rozpoczyna nową grę wywołując funkcję odNowa().
4. **Zmienne i rysowanie**
   - Definiuje zmienne pomocnicze i funkcję redraw(), która rysuje planszę i symbole graczy na ekranie.
5. **Główna pętla loop()**
   - Wczytuje wartość z analogowego pinu 3 i mapuje ją do wartości od 0 do 9, co odpowiada polom na planszy.
   - Sprawdza, czy nastąpiła zmiana wybranego pola na planszy.
   - W przypadku naciśnięcia przycisku, sprawdza czy pole jest wolne i aktualizuje planszę.
   - Po wykonaniu ruchu sprawdza, czy jest zwycięzca lub remis.
   - Wywołuje funkcję redraw() w przypadku ruchu lub początku nowej gry.
   - Po zakończeniu gry (zwycięstwo lub remis) resetuje planszę i czeka 1 sekundę przed rozpoczęciem nowej gry.

Kod



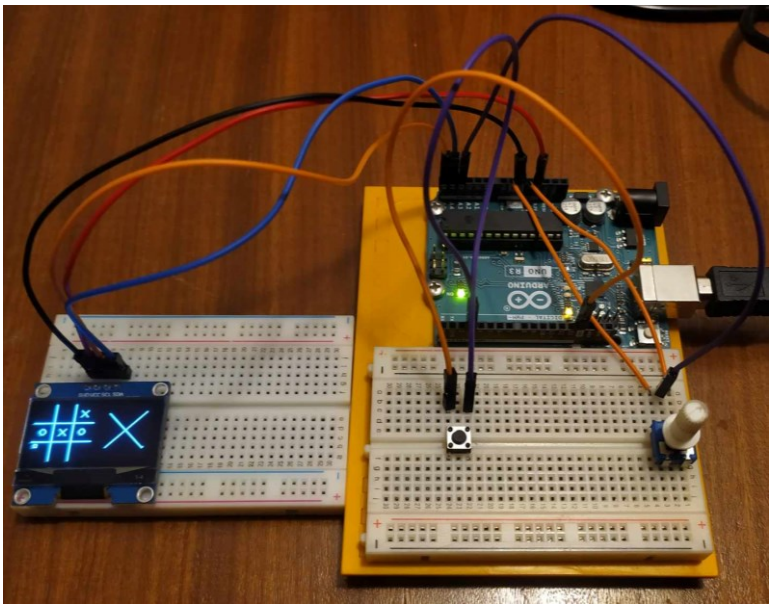

```
#include "lcdgfx.h"
DisplaySH1106_128x64_I2C display(-1);
NanoCanvas<128, 64, 1> canvas;
int plansza[3][3];
bool czyNowaGra = 1;
void odNowa() {
  for(int i=0;i<9;i++) plansza[i/3][i%3]=0;
  czyNowaGra = 1;
}
bool czyWygrana() {
  for (int i = 0; i < 3; i++) {
    if(plansza[i][0]!=0&&plansza[i][0]==plansza[i][1]&&plansza[i][0]==plansza[i][2]) return true;
    if(plansza[0][i]!=0&&plansza[0][i]==plansza[1][i]&&plansza[0][i]==plansza[2][i]) return true;
  }
  if(plansza[0][0]!=0&&plansza[0][0]==plansza[1][1]&&plansza[0][0]==plansza[2][2]) return true;
  if(plansza[0][2]!=0&&plansza[0][2]==plansza[1][1]&&plansza[0][2]==plansza[2][0]) return true;
  return false;
}
bool czyRemis() {
  for (int i = 0; i < 9; i++)
    if (plansza[i / 3][i % 3] == 0) return false;
  return true;
}
void setup() {
  pinMode(3, INPUT_PULLUP);
  display.begin();
  odNowa();
}
```

```
int index;
int X = 21;
int P = 7;
int gracz = 1;

void redraw() {
  canvas.clear();
  canvas.drawLine(0, X, X * 3, X);
  canvas.drawLine(0, 2 * X, X * 3, 2 * X);
  canvas.drawLine(X, 0, X, X * 3);
  canvas.drawLine(2 * X, 0, 2 * X, X * 3);
  for (int row = 0; row < 3; row++) {
    for (int col = 0; col < 3; col++) {
      if (plansza[row][col] == 1) {
        for (int i = 0; i < X - 2 * P + 1; i++) {
          canvas.putPixel(col * X + P + i, row * X + P + i);
          canvas.putPixel(col * X + P + i, (row + 1) * X - P - i);
        }
      } else if (plansza[row][col] == 2) {
        canvas.drawCircle(col * X + X / 2, row * X + X / 2, X / 2 - P);
      }
      if (index == row * 3 + col) {
        canvas.drawRect(col * X + 3, row * X + 3, col * X + 6, row * X + 6);
      }
    }
  }
  if (gracz == 1) {
    canvas.drawLine(80, 10, 120, 50);
    canvas.drawLine(80, 50, 120, 10);
  } else {
    canvas.drawCircle(100, 30, 20);
  }
  display.drawCanvas(0, 0, canvas);
}

int lastIndex = -1;

void loop() {
  int x = analogRead(3);
  index = map(x, 0, 1024, 0, 9);
  bool wygrana = false;
  bool bylruch = false;
  if (lastIndex != index) {
    lastIndex = index;
    bylruch = true;
  }
  if (digitalRead(3) == LOW) {
    bylruch = true;
    if (plansza[index / 3][index % 3] == 0) {
      plansza[index / 3][index % 3] = gracz;
      wygrana = czyWygrana();
      if (!wygrana) {
        gracz = gracz == 1 ? 2 : 1;
      }
    }
  }
  if (bylruch || czyNowaGra) redraw();
  czyNowaGra = false;
  if (wygrana || czyRemis()) {
    odNowa();
    delay(1000);
  }
  delay(50);
}
```



Możliwe ulepszenia

- Program można rozszerzyć o dodatkowe funkcje, takie jak wyświetlanie komunikatów o stanie gry (np. o wygranej lub remisie), co poprawiłoby doświadczenie użytkownika.
- Dodanie dźwięków lub animacji po zakończeniu gry mogłoby zwiększyć atrakcyjność wizualną gry.