

Hra Simon Says

Hra sa skladá zo štyroch farebných LED diód, ktoré sa postupne rozvedcujú a hráč musí tuto sekvenciu zopakovať v rovnakom poradí. Ak sa to hráčovi podarí sekvencia sa predĺži a tak sa stáva, každým kolom náročnejšia. Pokiaľ sa hráč pomýli znamená to pre neho prehru. Po stlačení tlačidla pripojeného ku kolíku A4 sa vygeneruje sekvencia a softvér čaká na rovnakú sekvenciu. Ak sú tieto dve sekvencie odlišné všetky LED diódy zablikajú dvakrát a, ak je sekvencia správna zablikajú raz.

Súčiastky:

Arduino UNO 1X

Vodiče 16X

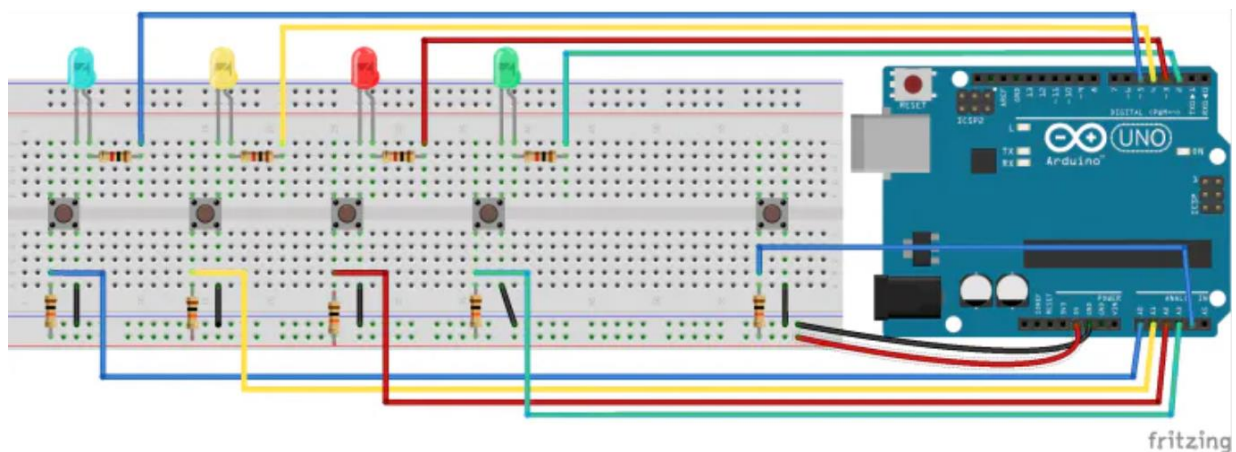
Breadboard 1X

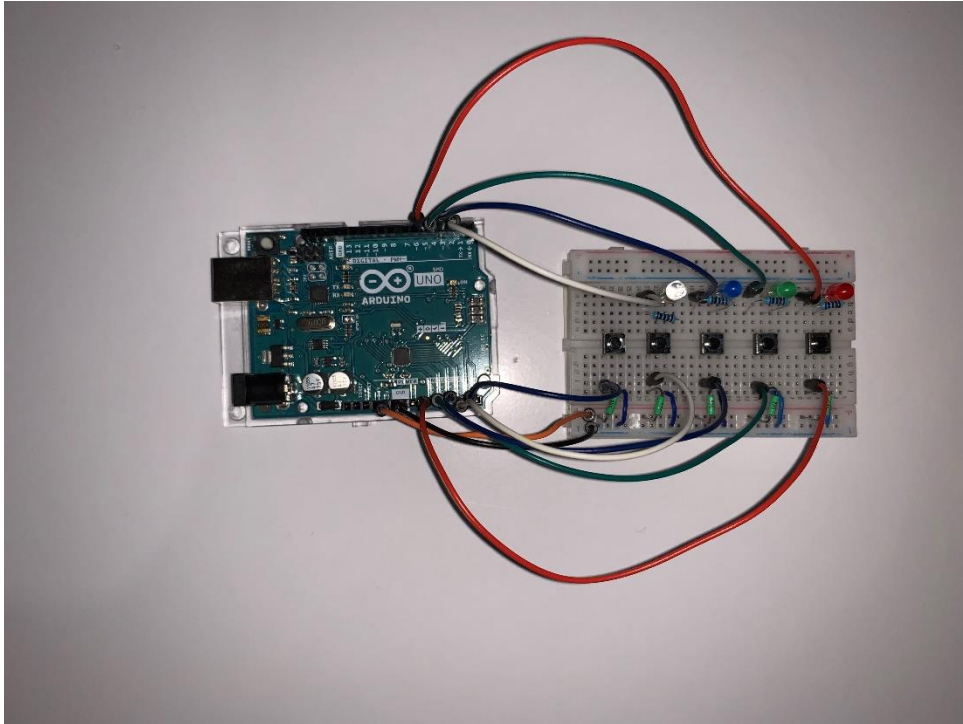
LED diódy 4X (biela, modrá, zelená, červená)

Rezistory 9X

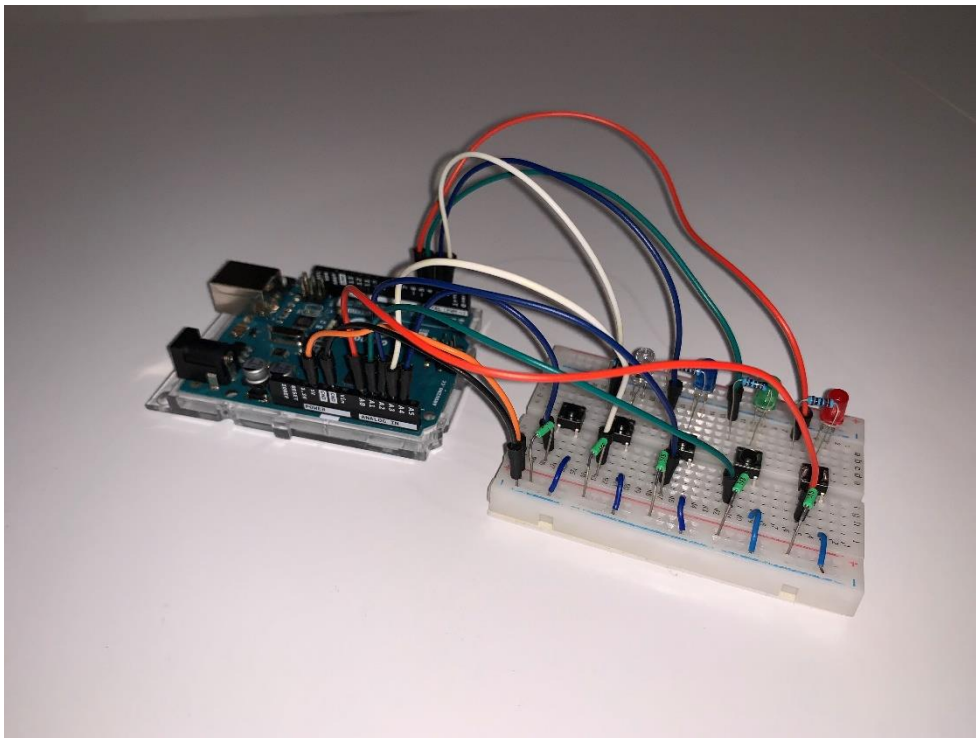
Tlačidlá 5X

Schéma:





Obr.1



Obr.2

Program:

```
const int MAX_LEVEL = 100;
int sequence[MAX_LEVEL];
int your_sequence[MAX_LEVEL];
```

```
int level = 1;
```

```
int velocity = 1000;
```

```
void setup() {  
  pinMode(A0, INPUT);  
  pinMode(A1, INPUT);  
  pinMode(A2, INPUT);  
  pinMode(A3, INPUT);
```

```
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);
```

```
  digitalWrite(2, LOW);  
  digitalWrite(3, LOW);  
  digitalWrite(4, LOW);  
  digitalWrite(5, LOW);  
}
```

```
void loop()  
{  
  if (level == 1)  
    generate_sequence();
```

```
  if (digitalRead(A4) == LOW || level != 1)  
  {  
    show_sequence();  
    get_sequence();  
  }  
}
```

```
void show_sequence()  
{  
  digitalWrite(2, LOW);  
  digitalWrite(3, LOW);  
  digitalWrite(4, LOW);  
  digitalWrite(5, LOW);
```

```
  for (int i = 0; i < level; i++)  
  {  
    digitalWrite(sequence[i], HIGH);  
    delay(velocity);
```

```
digitalWrite(sequence[i], LOW);  
delay(200);  
}  
}
```

```
void get_sequence()  
{  
int flag = 0;
```

```
for (int i = 0; i < level; i++)  
{  
flag = 0;  
while(flag == 0)  
{  
if (digitalRead(A0) == LOW)  
{  
digitalWrite(5, HIGH);  
your_sequence[i] = 5;  
flag = 1;  
delay(200);  
if (your_sequence[i] != sequence[i])  
{  
wrong_sequence();  
return;  
}  
digitalWrite(5, LOW);  
}  
}
```

```
if (digitalRead(A1) == LOW)  
{  
digitalWrite(4, HIGH);  
your_sequence[i] = 4;  
flag = 1;  
delay(200);  
if (your_sequence[i] != sequence[i])  
{  
wrong_sequence();  
return;  
}  
digitalWrite(4, LOW);  
}
```

```
if (digitalRead(A2) == LOW)  
{
```

```
digitalWrite(3, HIGH);
your_sequence[i] = 3;
flag = 1;
delay(200);
if (your_sequence[i] != sequence[i])
{
    wrong_sequence();
    return;
}
digitalWrite(3, LOW);
}
```

```
if (digitalRead(A3) == LOW)
{
    digitalWrite(2, HIGH);
    your_sequence[i] = 2;
    flag = 1;
    delay(200);
    if (your_sequence[i] != sequence[i])
    {
        wrong_sequence();
        return;
    }
    digitalWrite(2, LOW);
}
```

```
}
}
right_sequence();
}
```

```
void generate_sequence()
{
    randomSeed(millis());
```

```
for (int i = 0; i < MAX_LEVEL; i++)
{
    sequence[i] = random(2,6);
}
}
```

```
void wrong_sequence()
{
    for (int i = 0; i < 3; i++)
    {
```

```
digitalWrite(2, HIGH);  
digitalWrite(3, HIGH);  
digitalWrite(4, HIGH);  
digitalWrite(5, HIGH);  
delay(250);  
digitalWrite(2, LOW);  
digitalWrite(3, LOW);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
delay(250);  
}  
level = 1;  
velocity = 1000;  
}
```

```
void right_sequence()  
{  
digitalWrite(2, LOW);  
digitalWrite(3, LOW);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
delay(250);
```

```
digitalWrite(2, HIGH);  
digitalWrite(3, HIGH);  
digitalWrite(4, HIGH);  
digitalWrite(5, HIGH);  
delay(500);  
digitalWrite(2, LOW);  
digitalWrite(3, LOW);  
digitalWrite(4, LOW);  
digitalWrite(5, LOW);  
delay(500);
```

```
if (level < MAX_LEVEL);  
level++;
```

```
velocity -= 50;  
}
```