



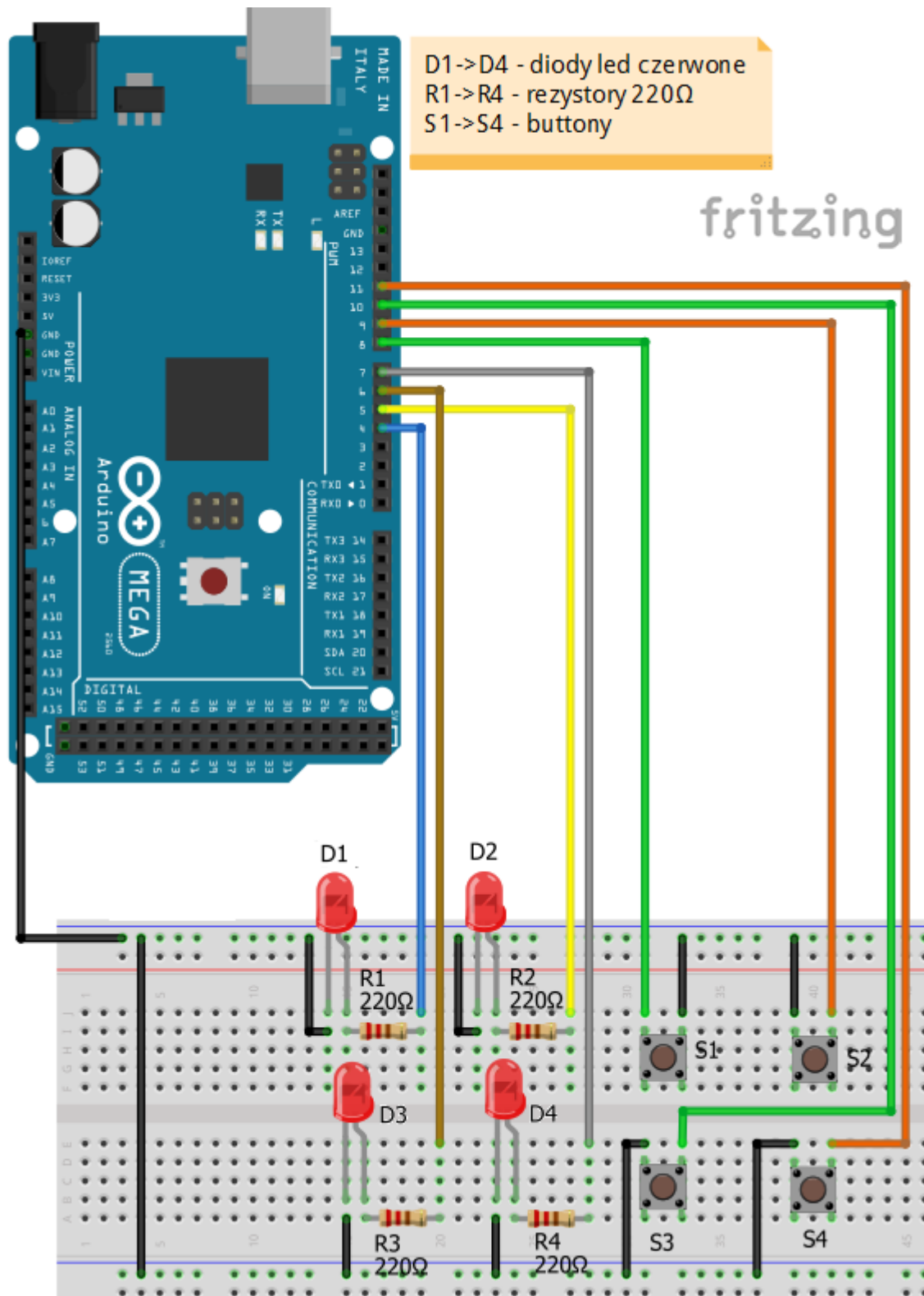
Nazwa implementacji:

Pamięć i zręczność - zapamiętaj kolejność

Autor:

Krzysztof Bytow

Opis implementacji: Budowa układu i programu do symulacji losowania jednej z sześciu liczb, jak w kostce do gry. Prezentacja wyniku losowania z wykorzystaniem diod elektroluminescencyjnych. Rozwijając wiedzę o zastosowaniu diod elektroluminescencyjnych, układ moduł-interfejsu zostaje rozbudowany o kolejne elementy. Efektem jest opracowanie implementacji pozwalającej ćwiczyć pamięć i zręczność.



Schemat połączeń:

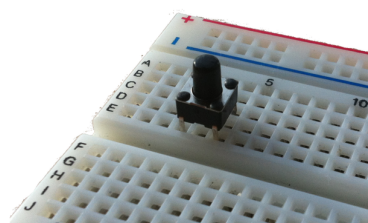
Uczeń/Uczennica po zestawieniu połączeń zgłasza nauczycielowi gotowość do sprawdzenia układu i wszystkich połączeń.



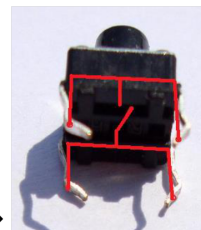
← dioda LED czerwona



oznaczenie kodem barwnym rezystora 220 Ω



← button →



Kod implementacji: Po złożeniu układu i jego oprogramowaniu, należy skupić się na diodach LED. Każda z nich ma przypisany odpowiedni button: przykładowo dioda D1 = S1, D2 = S2. Uruchomienie układu inicjuje losowanie zapalenia diody, należy następnie przycisnąć odpowiedni przycisk. Zostaje wylosowana dioda D4. Należy wybrać button, kolejny poziom to 2 diody wybrane losowo. Po zapaleniu należy powtórzyć sekwencję na przyciskach (zapalają nam się diody D2, D3, D3, D4, D1, D2) a po zgaśnięciu należy odtworzyć sekwencję na przyciskach S2, S3, S3, S4, S1, S2. Błędny wybór buttona oznajmiany jest zapaleniem wszystkich czterech diod.

```
int dioda1 = 4;           // przyznanie etykiety dla numeru pinu
int dioda2 = 5;           // przyznanie etykiety dla numeru pinu
int dioda3 = 6;           // przyznanie etykiety dla numeru pinu
int dioda4 = 7;           // przyznanie etykiety dla numeru pinu
int s1=8;                 // przyznanie etykiety dla numeru pinu
int s2=9;                 // przyznanie etykiety dla numeru pinu
int s3=10;                // przyznanie etykiety dla numeru pinu
int s4=11;                // przyznanie etykiety dla numeru pinu
int level=0;              // przypisanie wartości 0
int poziom[50];           // tworzymy deklarujemy tablicę
int u=0;                  // przypisanie wartości 0

void setup()              // początkowa konfiguracja - część przygotowująca układ do
{                          // działania
  Serial.begin(9600);      // ustawienie prędkości komunikacji
  for (int i=3; i<=7;i++)  // pętla for
  {
    pinMode(i,OUTPUT);     // ustawienie pinu jako wyjście
    digitalWrite(i,LOW);   // ustawienie stanu niskiego
  }
  pinMode(s1,INPUT);       // ustawienie pinu jako wejście
  digitalWrite(s1,HIGH);   // ustawienie stanu wysokiego
  pinMode(s2,INPUT);       // ustawienie pinu jako wejście
  digitalWrite(s2,HIGH);   // ustawienie stanu wysokiego
  pinMode(s3,INPUT);       // ustawienie pinu jako wejście
  digitalWrite(s3,HIGH);   // ustawienie stanu wysokiego
  pinMode(s4,INPUT);       // ustawienie pinu jako wejście
  digitalWrite(s4,HIGH);   // ustawienie stanu wysokiego
```



```
}  
  
void loop()           // główna pętla  
{  
  level=level++;     // zwiększ level  
  for (int z=0; z<=level;z++) // pętla for  
  {  
    int f=random(4,8); // przypisuje losową wartość z zakresu  
    digitalWrite(f,HIGH); // ustaw stan wysoki na wyjściach f  
    poziom[z]=f; // zapis do tablicy  
    delay(600); // czekaj 600 ms  
    digitalWrite(f,LOW); // ustaw stan niski  
    delay(300); // czekaj 300 ms  
  }  
  
  for (int p=0;p<=level;p++) // pętla for  
  {  
    int m=poziom[p]; // przypisanie wartości m  
    Serial.print("Dioda "); // wyświetl tekst  
    Serial.println(m); // wyświetl wartość m  
    u = button(); // przypisanie wartości z button  
    Serial.print("Wduszono przycisk numer "); // wyświetl tekst  
    Serial.println(u); // wyświetl wartość u  
    if (m == u) // instrukcja warunkowa  
    {  
      Serial.println("Wybrano poprawna diode"); // wyświetl tekst  
    }  
    else  
    {  
      Serial.println("Przegrana"); // wyświetl tekst  
      off(); // skocz do funkcji off  
    }  
  }  
}  
  
void off()           // funkcja ogg zapala i gasi diody  
{  
  digitalWrite(4,HIGH); // ustawienie stanu wysokiego  
  digitalWrite(5,HIGH); // ustawienie stanu wysokiego  
  digitalWrite(6,HIGH); // ustawienie stanu wysokiego  
  digitalWrite(7,HIGH); // ustawienie stanu wysokiego  
  delay(5000); //czekaj 5 sekund  
  digitalWrite(4,LOW); // ustawienie stanu niskiego  
  digitalWrite(5,LOW); // ustawienie stanu niskiego  
  digitalWrite(6,LOW); // ustawienie stanu niskiego  
  digitalWrite(7,LOW); // ustawienie stanu niskiego  
}  
  
boolean get(int e)  
{  
  switch(e) // warunek wielokrotnego wyboru  
  {  
    case 4:  
      return digitalRead(s1) == LOW;
```



```
case 5:  
return digitalRead(s2) == LOW;
```

```
case 6:  
return digitalRead(s3) == LOW;
```

```
case 7:  
return digitalRead(s4) == LOW;
```

```
default:  
return false;
```

```
}  
}  
int button()  
{  
int a = 0;           // przypisz wartość  
while(!a)          // pętla while  
{  
if(get(4))          // instrukcja warunkowa  
a = 4;              // przypisz wartość  
if(get(5))          // instrukcja warunkowa  
a = 5;              // przypisz wartość  
if(get(6))          // instrukcja warunkowa  
a = 6;              // przypisz wartość  
if(get(7))          // instrukcja warunkowa  
a = 7;              // przypisz wartość  
}  
delay(180);         // czekaj 180ms  
return a;           // zwraca wartość a  
}
```

Film instruktażowy: <http://youtu.be/evllq2Zpwgl>

