Zadanie 1: dioda LED

- Cel zadania:
  - Mamy sprawić aby dwie diody mrugały naprzemiennie

Zadanie 1: budowa algorytmu

1. Ustalenie wejść / wyjść układu
2. Inicjalizacja wejść / wyjść układu
3. Budowa głównej pętli
    1. Zapal pierwszą diodę i zgaś drugą diodę
    2. Odczekaj trochę czasu
    3. Zgaś pierwszą diodę i zapal drugą diodę
    4. Odczekaj trochę czasu

Zadanie 1: gotowe rozwiązanie

```
#include <Arduino.h>

int LED_H_PIN = 4;
int LED_V_PIN = 7;

void setup(){
    pinMode(LED_H_PIN, OUTPUT);
    pinMode(LED_V_PIN, OUTPUT);
}
int isH = true;

void loop(){
    if(isH){
        digitalWrite(LED_H_PIN, HIGH);
        digitalWrite(LED_V_PIN, LOW);
    }
    else{
        digitalWrite(LED_V_PIN, HIGH);
        digitalWrite(LED_H_PIN, LOW);
    }
    isH = !isH;
    delay(50);
}
```

Ustalenie wejść / wyjść układu

Inicjalizacja wejść / wyjść układu

Główna pętla programu

Zapal pierwszą diodę i zgaś drugą diodę

Zapal drugą diodę i zgaś pierwszą diodę

Odczekaj trochę czasu

Zadanie 2: dioda LED z przyciskiem

- Cel zadania:
  - Mamy sprawić, aby diody zmieniły swój stan po naciśnięciu przycisku

Zadanie 2: Budowa algorytmu

1. Ustalenie wejść / wyjść układu
2. Inicjalizacja diod
3. Inicjalizacja przycisku
4. Inicjalizacja zmiennych
5. Budowa głównej pętli
   1. Odczyt stanu przycisku
   2. Zapal pierwszą diodę i zgaś drugą diodę
   3. Odczekaj trochę czasu
   4. Zgaś pierwszą diodę i zapal drugą diodę
   5. Odczekaj trochę czasu

Zadanie 2: gotowe rozwiązanie

```
#include <Arduino.h>

int LED_H_PIN = 4;
int LED_V_PIN = 7;
int BUTTON_PIN = 2;

void setup()
{
    pinMode(LED_H_PIN, OUTPUT);
    pinMode(LED_V_PIN, OUTPUT);

    pinMode(BUTTON_PIN, INPUT);

    Serial.begin(9600);
}

int isH = true;
int wasButtonPressed = LOW;
```

Ustalenie wejść / wyjść układu

Inicjalizacja diod

Inicjalizacja przycisku

Inicjalizacja zmiennych

Zadanie 2: gotowe rozwiązanie

```
void loop(){
int buttonPressed = digitalRead(BUTTON_PIN);

    Serial.print(" | button: " );
    Serial.print(buttonPressed);

    if (wasButtonPressed == LOW && buttonPressed == HIGH){
        isH = !isH;
    }
    wasButtonPressed = buttonPressed;
    if(isH){
        digitalWrite(LED_H_PIN, HIGH);
        digitalWrite(LED_V_PIN, LOW);
    }
    else{
        digitalWrite(LED_V_PIN, HIGH);
        digitalWrite(LED_H_PIN, LOW);
    }
    Serial.println();
    delay(50);
}
```

Odczyt stanu przycisku

Zapal pierwszą diodę i zgaś drugą diodę

Zapal drugą diodę i zgaś pierwszą diodę

Odczekaj trochę czasu

Zadanie 3: diody i potencjometr

- Cel zadania:
  - Mamy sprawić, aby diody zmieniły swój stan po naciśnięciu przycisku (podobnie jak w zadaniu 2)
  - Mamy za zadanie odczytać stan wejścia analogowego połączonego z potencjometrem

Zadanie 3: Budowa algorytmu

1. Ustalenie wejść / wyjść układu
2. Inicjalizacja diod
3. Inicjalizacja przycisku
4. Inicjalizacja wejścia analogowego
5. Inicjalizacja zmiennych
6. Budowa głównej pętli
   1. Odczyt stanu przycisku
   2. Odczytaj stan wejścia analogowego
   3. Zapal pierwszą diodę i zgaś drugą diodę
   4. Odczekaj trochę czasu
   5. Zgaś pierwszą diodę i zapal drugą diodę
   6. Odczekaj trochę czasu

Zadanie 3: gotowe rozwiązanie

```
#include <Arduino.h>

int LED_H_PIN = 4;
int LED_V_PIN = 7;
int  BUTTON_PIN = 2;

int POTENTIOMETR_PIN = A0;

int POTENTIOMETR_LEVEL  = 3;

void setup(){
    pinMode(LED_H_PIN, OUTPUT);
    pinMode(LED_V_PIN, OUTPUT);

    pinMode(BUTTON_PIN, INPUT);

    Serial.begin(9600);
}
int isH = true;
int wasButtonPressed = LOW;
```

Ustalenie wejść / wyjść układu

Inicjalizacja diod

Inicjalizacja przycisku

Inicjalizacja wejścia analogowego

Inicjalizacja zmiennych

Zadanie 3: gotowe rozwiązanie

```
void loop(){
    int buttonPressed = digitalRead(BUTTON_PIN);

    Serial.print(" | button: " );
    Serial.print(buttonPressed);

    if (wasButtonPressed == LOW && buttonPressed == HIGH){
        isH = !isH;
    }
    wasButtonPressed = buttonPressed;

    if(isH){
        digitalWrite(LED_H_PIN, HIGH);
        digitalWrite(LED_V_PIN, LOW);
    }
    else{
        digitalWrite(LED_V_PIN, HIGH);
        digitalWrite(LED_H_PIN, LOW);
    }
     int potentiometrValue = analogRead(POTENTIOMETR_PIN);
     int mappedPotentiometrValue = map(potentiometrValue, 0, 1024, 1, POTENTIOMETR_LEVEL*

     Serial.print(" | potentiometr: " );
     Serial.print(potentiometrValue);
     Serial.print(" | mapped: ");
     Serial.print(mappedPotentiometrValue);
     Serial.println();
     delay(50);
}
```

Odczyt stanu przycisku

Zapal pierwszą diodę i zgaś drugą diodę

Zapal drugą diodę i zgaś pierwszą diodę

Odczytaj stan wejścia analogowego

Wyświetl stan wejścia analogowego

Odczekaj trochę czasu

Zadanie 4: panel słoneczny 1

- Cel zadania:
  - Mamy zbudować platformę dla panelu słonecznego sterowanego 2 silnikami.
  - Panel ma byś sterowany ręcznie, przy użyciu potencjometru

Zadanie 4: Budowa algorytmu

1. Ustalenie wejść / wyjść układu
2. Inicjalizacja diod
3. Inicjalizacja przycisku
4. Inicjalizacja wejścia analogowego
5. Inicjalizacja zmiennych
6. Budowa głównej pętli
   1. Odczyt stanu przycisku
   2. Odczytaj stan wejścia analogowego
   3. Porusz silnikami zgodnie z odczytaną wartością
   4. Zapal pierwszą diodę i zgaś drugą diodę
   5. Odczekaj trochę czasu
   6. Zgaś pierwszą diodę i zapal drugą diodę
   7. Odczekaj trochę czasu

Zadanie 4: gotowe rozwiązanie

```
#include <Arduino.h>
#include <Servo.h>

int LED_H_PIN = 4;
int LED_V_PIN = 7;
int BUTTON_PIN = 2;
int POTENTIOMETR_PIN = A0;
int POTENTIOMETR_LEVEL  = 3;

int SERVO_H_PIN = 9;
int SERVO_V_PIN = 10;
Servo serverH;
Servo serverV;

void setup(){
    pinMode(LED_H_PIN, OUTPUT);
    pinMode(LED_V_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT);
    serverH.attach(SERVO_H_PIN);
    serverV.attach(SERVO_V_PIN);

    Serial.begin(9600);
}
int isH = true;
int wasButtonPressed = LOW;
```

Ustalenie wejść / wyjść układu

Inicjalizacja diod

Inicjalizacja przycisku

Inicjalizacja silników

Inicjalizacja zmiennych

Zadanie 4: gotowe rozwiązanie

```
void loop(){
    int buttonPressed = digitalRead(BUTTON_PIN);
    Serial.print(" | button: " );
    Serial.print(buttonPressed);

    if (wasButtonPressed == LOW && buttonPressed == HIGH){
        isH = !isH;
    }
    wasButtonPressed = buttonPressed;

    if(isH){
        digitalWrite(LED_H_PIN, HIGH);
        digitalWrite(LED_V_PIN, LOW);
    }
    else{
        digitalWrite(LED_V_PIN, HIGH);
        digitalWrite(LED_H_PIN, LOW);
    }

    int potentiometrValue = analogRead(POTENTIOMETR_PIN);
    int mappedPotentiometrValue = map(potentiometrValue, 0, 1024, 1, POTENTIOMETR_LEVEL*2);

    Serial.print(" | potentiometr: " );
    Serial.print(potentiometrValue);
    Serial.print(" | mapped: ");
    Serial.print(mappedPotentiometrValue);

    ...
```

Odczyt stanu przycisku

Zapal pierwszą diodę i zgaś drugą diodę

Zapal drugą diodę i zgaś pierwszą diodę

Odczytaj stan wejścia analogowego

Wyświetl stan wejścia analogowego

Zadanie 4: gotowe rozwiązanie

```
if (mappedPotentiometrValue != POTENTIOMETR_LEVEL){

        int delta = mappedPotentiometrValue - POTENTIOMETR_LEVEL;
        if (isH){
            serverH.write(serverH.read() + delta);
        }
        else{
            serverV.write(serverV.read() + delta);
        }

        digitalWrite(LED_V_PIN, LOW);
        digitalWrite(LED_H_PIN, LOW);
    }
    Serial.print("| servo H: " );
    Serial.print(serverH.read());
    Serial.print("| server V: ");
    Serial.print(serverV.read());

    Serial.println();
    delay(50);
}
```

Odczytaj zmianę stanu przycisku

Ustaw odpowiednie wartości do silników

Wyświetl stan silników

Zadanie 5: panel słoneczny 2

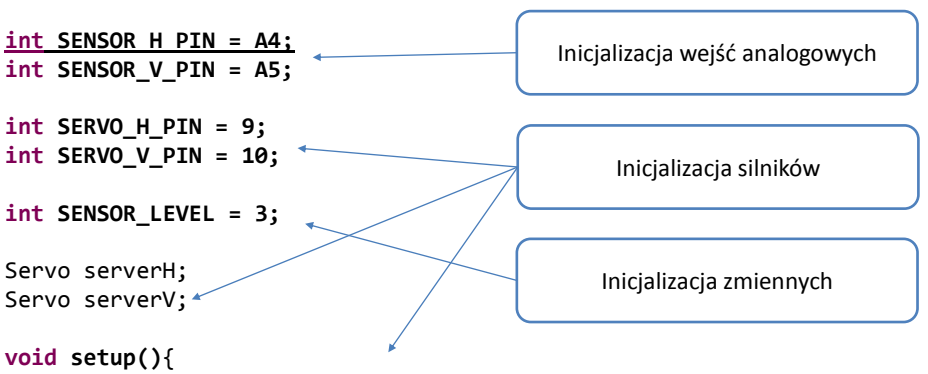
- Cel zadania:
  - Mamy zbudować platformę dla panelu słonecznego sterowanego 2 silnikami.
  - Panel ma byś sterowany automatycznie i podążać za światłem, mierzonym poprzez czujnik natężenia światła

Zadanie 4: Budowa algorytmu

1. Ustalenie wejść / wyjść układu
2. Inicjalizacja wejść analogowych
3. Inicjalizacja zmiennych
4. Budowa głównej pętli
    1. Odczyt stanu pierwszego czujnika
    2. Odczyt stanu drugiego silnika
    3. Poruszenie pierwszym silnikiem
    4. Poruszenie drugim silnikiem

Zadanie 5: gotowe rozwiązanie

```
#include <Arduino.h>
#include <Servo.h>

int SENSOR_H_PIN = A4;
int SENSOR_V_PIN = A5;

int SERVO_H_PIN = 9;
int SERVO_V_PIN = 10;

int SENSOR_LEVEL = 3;

Servo serverH;
Servo serverV;

void setup(){
    serverH.attach(SERVO_H_PIN);
    serverV.attach(SERVO_V_PIN);
    Serial.begin(9600);
}
```



Zadanie 5: gotowe rozwiązanie

```
void loop(){
    int sensorH = analogRead(SENSOR_H_PIN);
    int mappedSensorH = map(sensorH, 0, 1024, 1, SENSOR_LEVEL*2);

    int sensorV = analogRead(SENSOR_V_PIN);
    int mappedSensorV = map(sensorV, 0, 1024, 1, SENSOR_LEVEL*2);

    Serial.print(" | sensorH: " );
    Serial.print(sensorH);
    Serial.print(" | ");
    Serial.print(mappedSensorH);

    Serial.print(" | sensorV: " );
    Serial.print(sensorV);
    Serial.print(" | ");
    Serial.print(mappedSensorV);
...
```

Odczytaj dane z sensorów H i V

Wyświetl odczytane dane

Zadanie 5: gotowe rozwiązanie

```
 if (mappedSensorH != SENSOR_LEVEL){
        int delta = mappedSensorH - SENSOR_LEVEL;
        serverH.write(serverH.read() + delta);
    }

    if (mappedSensorV != SENSOR_LEVEL){
        int delta = mappedSensorV - SENSOR_LEVEL;
        serverV.write(serverV.read() + delta);
    }

    Serial.print("| servo H: " );
    Serial.print(serverH.read());
    Serial.print("| server V: ");
    Serial.print(serverV.read());

    Serial.println();
    delay(50);
}
```

Sprawdź czy jest konieczne poruszanie silnikiem 1 i porusz nim

Sprawdź czy jest konieczne poruszanie silnikiem 2 i porusz nim

Wyświetl dane z czujników

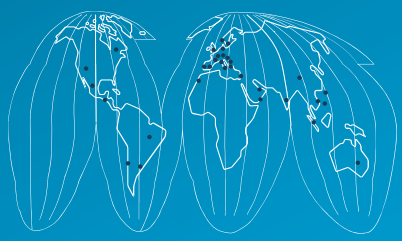
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 130,000 people in 44 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2012 global revenues of EUR 10.3 billion.
Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Rightshore® is a trademark belonging to Capgemini



www.capgemini.com