|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SPRAWOZDANIE** | | | | | | **PROSZĘ PODAĆ NR GRUPY:** | | | | | | | |
| **ZIISS1** | | | **3** | **5** | **1** | **2** | **IO** |
| **IMIĘ** | **NAZWISKO** | **Temat ćwiczenia zgodny z wykazem tematów:** | **PONIŻEJ PROSZĘ PODAĆ TERMIN ZAJĘĆ:** | | | | | **ROK:** | | | | | |
| **Podstawy programowania Układy wejścia-wyjścia** | **2023 r.** | | | | | |
| **MICHAŁ** | **WARSZAWSKI** | **PN** | **WT** | **SR** | | **CZ** | | **PT** | | **SB** | | **ND** |
| **GODZINA ROZPOCZĘCIA ZAJĘĆ:** | | | | | | | | **11 : 30** | | |
|  | | | | | | | | | | | | | |

Wprowadzenie teoretyczne:

**Opisz rodzaje portów w mikrokontrolerze ATmega2560**

Mikrokontroler ATmega2560, produkowany przez firmę Microchip, jest wyposażony w różne rodzaje portów, które pełnią różne funkcje.

* Porty ogólne (General Purpose I/O Ports) obejmują PORTA, PORTB, PORTC, PORTD, PORTE, PORTF, PORTG, PORTH, PORTJ i PORTK. Każdy z tych portów składa się z 8 pinów, które mogą być skonfigurowane jako wejścia lub wyjścia cyfrowe.
* Porty wejścia/wyjścia (I/O Ports) pozwalają na indywidualną konfigurację pinów jako wejścia lub wyjścia cyfrowe.
* Porty analogowe, takie jak A0, A1, ..., A15, umożliwiają korzystanie z przetworników analogowo-cyfrowych (ADC) do pomiaru napięć.
* Porty szeregowe (Serial Ports) obejmują interfejsy UART, SPI i I2C do komunikacji z innymi urządzeniami.
* Porty PWM (Pulse Width Modulation) umożliwiają generowanie sygnałów PWM do sterowania prędkością silników, jasnością diod LED, itp.
* Porty zegarowe, takie jak XTAL1 i XTAL2, służą do podłączania rezonatora kwarcowego lub oscylatora zegarowego.
* Porty wejścia przerwania (Interrupt Ports) pozwalają na natychmiastowe przerwanie programu w odpowiedzi na zmiany stanu na danym pinie.

**Zadanie 1**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Listing programu:**

*#define people\_red A4*

*#define people\_green A0*

*#define cars\_red A15*

*#define cars\_yellow A12*

*#define cars\_green A9*

*void setup() {*

*pinMode(cars\_red, OUTPUT);*

*pinMode(cars\_yellow, OUTPUT);*

*pinMode(cars\_green, OUTPUT);*

*pinMode(people\_red, OUTPUT);*

*pinMode(people\_green, OUTPUT);*

*// TEST*

*// zapalenie wszystkich świateł*

*digitalWrite(cars\_red, HIGH);*

*digitalWrite(cars\_yellow, HIGH);*

*digitalWrite(cars\_green, HIGH);*

*digitalWrite(people\_red, HIGH);*

*digitalWrite(people\_green, HIGH);*

*delay(1000);*

*// zgaszenie wszystkich świateł*

*digitalWrite(cars\_red, LOW);*

*digitalWrite(cars\_yellow, LOW);*

*digitalWrite(cars\_green, LOW);*

*digitalWrite(people\_red, LOW);*

*digitalWrite(people\_green, LOW);*

*delay(1000);*

*}*

*int loops = 0;*

*void loop(){*

*loops++;*

*if(loops<=2){*

*// TRYB DZIENNY*

*digitalWrite(cars\_green, HIGH);*

*digitalWrite(people\_red, HIGH);*

*delay(10000);*

*digitalWrite(cars\_green, LOW);*

*digitalWrite(cars\_yellow, HIGH);*

*delay(2000);*

*digitalWrite(cars\_yellow, LOW);*

*digitalWrite(cars\_red, HIGH);*

*delay(4000);*

*digitalWrite(people\_red, LOW);*

*digitalWrite(people\_green, HIGH);*

*delay(5000);*

*for(int i=0; i<5; i++){*

*digitalWrite(people\_green, HIGH);*

*delay(500);*

*digitalWrite(people\_green, LOW);*

*delay(500);*

*}*

*digitalWrite(people\_red, HIGH);*

*delay(2000);*

*digitalWrite(cars\_red, LOW);*

*digitalWrite(cars\_yellow, HIGH);*

*delay(1000);*

*digitalWrite(cars\_yellow, LOW);*

*digitalWrite(people\_red, LOW);*

*}*

*else{*

*// TRYB NOCNY*

*digitalWrite(cars\_yellow, HIGH);*

*delay(500);*

*digitalWrite(cars\_yellow, LOW);*

*delay(500);*

*}*

*}*

**Zrzut ekranu ze schematem:**

Obraz zawierający Prostokąt, Równolegle, zrzut ekranu, linia

Opis wygenerowany automatycznie

**Wnioski:**

W kodzie definiowane są stałe dla konkretnych pinów, ułatwiając tym samym czytelność. Następnie w funkcji setup() ustawiane są piny jako wyjścia dla świateł samochodów i pieszych. Sekcja testowa sprawdza poprawność działania świateł. W pętli głównej programu loop(), jest zaimplementowana sekwencja świetlna dla trybu dziennego i nocnego. W trybie dziennym symulowana jest typowa sekwencję świateł na skrzyżowaniu, a także efekt migotania świateł dla pieszych. Po dwóch obiegach pętli, program przechodzi w tryb nocny, gdzie jedynie migają światła żółte samochodów.

**Zadanie 2**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Listing programu:**

#define people\_red A4

#define people\_green A0

#define cars\_red A15

#define cars\_yellow A12

#define cars\_green A9

void setup(){

pinMode(cars\_red, OUTPUT);

pinMode(cars\_yellow, OUTPUT);

pinMode(cars\_green, OUTPUT);

pinMode(people\_red, OUTPUT);

pinMode(people\_green, OUTPUT);

// TEST

turnOnOffAll(HIGH);

delay(1000);

turnOnOffAll(LOW);

delay(1000);

// PRZERWANIE

pinMode(13, INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(13), nightMode, FALLING);

interrupts();

}

void loop(){

if(digitalRead(13)==1){

// TRYB DZIENNY

digitalWrite(cars\_green, HIGH);

digitalWrite(people\_red, HIGH);

delay(10000);

digitalWrite(cars\_green, LOW);

digitalWrite(cars\_yellow, HIGH);

delay(2000);

digitalWrite(cars\_yellow, LOW);

digitalWrite(cars\_red, HIGH);

delay(4000);

digitalWrite(people\_red, LOW);

digitalWrite(people\_green, HIGH);

delay(5000);

for(int i=0; i<5; i++){

digitalWrite(A0, HIGH);

delay(500);

digitalWrite(A0, LOW);

delay(500);

}

digitalWrite(people\_red, HIGH);

delay(2000);

digitalWrite(cars\_red, LOW);

digitalWrite(cars\_yellow, HIGH);

delay(1000);

digitalWrite(cars\_yellow, LOW);

digitalWrite(people\_red, LOW);

}

else{  
 // TRYB NOCNY

digitalWrite(cars\_yellow, HIGH);

delay(500);

digitalWrite(cars\_yellow, LOW);

delay(500);

}

}

void nightMode(){

return 0;

}

void turnOnOffAll(int state){

digitalWrite(cars\_red, state);

digitalWrite(cars\_yellow, state);

digitalWrite(cars\_green, state);

digitalWrite(people\_red, state);

digitalWrite(people\_green, state);

}

**Zrzut ekranu ze schematem:**

Obraz zawierający Prostokąt, Równolegle, linia, design

Opis wygenerowany automatycznie

**Wnioski:**

W przedstawionym kodzie mikrokontrolera dodatkowo skonfigurowano przerwanie na pinie 13, aby obsłużyć tryb nocny. W pętli `loop()` sprawdzany jest stan przycisku (pin 13). Sekwencja dla trybu dziennego jest wykonywana dopóki nie zostanie włączony przycisk. Wtedy aktywowana jest sekwencja nocna i trwa dopóki przycisk nie zostanie wyłączony.

**Zadanie 3**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Listing programu:**

*#define PEOPLE\_RED A4*

*#define PEOPLE\_GREEN A0*

*#define CARS\_RED A15*

*#define CARS\_YELLOW A12*

*#define CARS\_GREEN A9*

*#define NIGHT\_ON\_PIN 13*

*#define BUTTON\_PIN 1*

*#define LED\_PIN 6*

*void setup() {*

*pinMode(CARS\_RED, OUTPUT);*

*pinMode(CARS\_YELLOW, OUTPUT);*

*pinMode(CARS\_GREEN, OUTPUT);*

*pinMode(PEOPLE\_RED, OUTPUT);*

*pinMode(PEOPLE\_GREEN, OUTPUT);  
pinMode(LED\_PIN, OUTPUT);*

*// TEST*

*turnOnOffAll(HIGH);*

*delay(1000);*

*turnOnOffAll(LOW);*

*delay(1000);*

*// PRZERWANIA*

*pinMode(NIGHT\_ON\_PIN, INPUT\_PULLUP);*

*attachInterrupt(digitalPinToInterrupt(NIGHT\_ON\_PIN), przerwanie, FALLING);*

*pinMode(BUTTON\_PIN, INPUT);*

*attachInterrupt(digitalPinToInterrupt(BUTTON\_PIN), przerwanie, FALLING);*

*interrupts();*

*}*

*void loop() {*

*// jeżeli tryb nocny ON*

*if(digitalRead(NIGHT\_ON\_PIN) == LOW){*

*digitalWrite(CARS\_GREEN, LOW);*

*digitalWrite(PEOPLE\_RED, LOW);*

*digitalWrite(CARS\_YELLOW, HIGH);*

*delay(500);*

*digitalWrite(CARS\_YELLOW, LOW);*

*delay(500);*

*}*

*// jeżeli tryb nocny OFF*

*else{*

*// kliknięto przycisk pieszych*

*if((digitalRead(BUTTON\_PIN) == LOW)){*

*digitalWrite(LED\_PIN, HIGH);*

*delay(2000);*

*digitalWrite(CARS\_GREEN, LOW);*

*digitalWrite(CARS\_YELLOW, HIGH);*

*delay(2000);  
digitalWrite(CARS\_YELLOW, LOW);*

*digitalWrite(CARS\_RED, HIGH);*

*delay(2000);*

*digitalWrite(PEOPLE\_RED, LOW);*

*digitalWrite(PEOPLE\_GREEN, HIGH);*

*digitalWrite(LED\_PIN, LOW);*

*delay(5000);*

*for(int i=0; i<5; i++){*

*digitalWrite(PEOPLE\_GREEN, HIGH);*

*delay(500);*

*digitalWrite(PEOPLE\_GREEN, LOW);*

*delay(500);*

*}*

*digitalWrite(PEOPLE\_RED, HIGH);*

*delay(2000);*

*digitalWrite(CARS\_RED, LOW);*

*digitalWrite(CARS\_YELLOW, HIGH);*

*delay(1000);*

*digitalWrite(CARS\_YELLOW, LOW);*

*}*

*// przycisk nie aktywny*

*else{*

*digitalWrite(CARS\_YELLOW, LOW);*

*digitalWrite(CARS\_GREEN, HIGH);*

*digitalWrite(PEOPLE\_RED, HIGH);*

*}*

*}*

*}*

*void przerwanie() {*

*return;*

*}*

*void turnOnOffAll(int state) {*

*digitalWrite(CARS\_RED, state);*

*digitalWrite(CARS\_YELLOW, state);*

*digitalWrite(CARS\_GREEN, state);*

*digitalWrite(PEOPLE\_RED, state);*

*digitalWrite(PEOPLE\_GREEN, state);*

*}*

**Zrzut ekranu ze schematem:**

**Obraz zawierający zrzut ekranu, linia, Równolegle, Prostokąt

Opis wygenerowany automatycznie**

**Wnioski:**

W przedstawionym kodzie mikrokontrolera dodatkowo skonfigurowano przerwanie na pinie 1, aby obsłużyć wymuszanie świateł dla pieszych. W pętli `loop()` sprawdzany jest stan przycisków (pin 13 i pin 1). Światła zielone dla samochodów i czerwone dla pieszych są zapalone dopóki nie zostanie kliknięty przycisk pieszych. Wtedy aktywowana jest sekwencja dla pieszych a po jej zakończeniu ponownie zielone światło otrzymują samochody. Równolegle aktywna jest obsługa przycisku dla trybu nocnego, który ma pierwszeństwo nad oba rodzajami schematów dziennych. Przyciski dla pieszych w trybie nocnym są nieaktywne.