

LABORATORIUM 1

Obsługa cyfrowych urządzeń we/wy

1. Cel ćwiczenia.

Celem ćwiczenia jest praktyczne zrozumienie koncepcji obsługi cyfrowych urządzeń wejścia/wyjścia. W ramach ćwiczenia, skonfigurujesz pin mikrokontrolera jako cyfrowe wejście/wyjście. Nauczysz się sprawdzać stan portu urządzenia cyfrowego oraz ustawiać poziom napięcia na wyjściu urządzenia cyfrowego.

2. Zadania do wykonania.

Zadanie 1. Wyjścia cyfrowe:

Krok 1: Podłączenie multimetru do pinu Arduino

- Podłącz przewód do pinu D10 Arduino oraz drugi przewód do pinu GND Arduino (masa)
- Połącz przewód dodatni (czerwony) multimetru ze złączem D10, przewód ujemny (czarny) z masą (GND)
- Ustaw pokrętką tryb pomiaru napięcia stałego w zakresie do 20V

Krok 2: Program Arduino.

- W oprogramowaniu Arduino, napisz poniższy przykładowy program

```
const int outPin = 10;

void setup() {
  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
  Serial.println("Wybierz 0 lub 1");
}

void loop() {
  if (Serial.available() > 0) {
    char c = Serial.read();
    if (c == '1')
      digitalWrite(outPin, HIGH);
    else if (c == '0');
    digitalWrite(outPin, LOW);
  }
}
```

- Zaprogramuj Arduino i uruchom monitor portu szeregowego: **Narzędzia -> Monitor portu szeregowego**
- Po podłączeniu multimetru do Arduino powinieneś widzieć, jak po przesłaniu do Arduino za pośrednictwem monitora portu szeregowego poleceń „0” lub „1”, napięcie na złączu zmienia się z 0V do około 5V

Krok 3: Sygnalizacja za pomocą LED.

- Zamiast multimetru, podłącz do pinu D10 diodę LED. **Pamiętaj o rezystorze ograniczającym prąd diody!**

Zadanie 2. Wejścia cyfrowe:

Krok 1: Skonfiguruj pin D5 jako wejście cyfrowe.

- Uruchom nowy projekt
- Do funkcji **Setup()** dodaj następującą komendę

```
pinMode(inputPin, INPUT);|
```

- Nie zapomnij o wcześniejszym dodaniu odpowiedniej deklaracji stałej *inputPin*

```
const int inputPin = 5;|
```

Krok 2: Odczyt stanu pinu cyfrowego.

- Dodaj do funkcji **loop()** następujący kod (pamiętaj o wcześniejszym ustawieniu szybkości transmisji UART) przy pomocy którego, odczytywany będzie stan pinu D5 w odstępach co 1 s

```
int reading = digitalRead(inputPin);  
Serial.println(reading);  
delay(1000);
```

- Uruchom monitor portu szeregowego i sprawdź odczytywaną wartości, jednocześnie naprzemiennie ściskając i popuszczając palcami koniec przewodu. Dlaczego Arduino odczytuje „0” i „1”?

Krok 3: Podłączenie przycisku

- Do pinu D5 podłącz przycisk (microswitch) zgodnie ze schematem podanym przez prowadzącego
- Sprawdź efekt naciśnięcia przycisku na wyświetlaną w monitorze szeregowym wartość zależnie od przyjętej konfiguracji pinu wejściowego

Krok 4: Zadanie do samodzielnego wykonania

- Wykorzystując kod z zadania 1 oraz 2, napisz program przy pomocy którego po wciśnięciu przycisku zapalana będzie dioda LED, a w przypadku nie naciśnięcia przycisku, LED pozostanie „zgaszony”

Zadanie 3. Eliminacja drgań styków:

Krok 1: Obserwacja zjawiska drgania zestyków przełącznika

- podłącz sondę oscyloskopu do złącza przycisku
- Zaobserwuj na ekranie oscyloskopu efekt naprzemiennego naciskania przycisku. Co można zaobserwować? Jaka jest tego przyczyna?

Krok 2: Eliminacja programowa drgań styków

- umieść w programie funkcję opóźniającą przy odczytywaniu stanu wejścia cyfrowego zgodnie z instrukcją prowadzącego