1. Mruganie diodą
   1. **Cel zadania:** Zadanie polega na poznaniu podstawowych elementów programu pisanego dla mikrokontrolerów AtMega16 w platformie Arduino oraz konstrukcji najprostszego układu elektrycznego złożonego z diody i rezystora.
   2. **Niezbędne kroki:**
      1. Konstrukcja poniższego układu elektrycznego:

<Schemat elektryczny>

* + 1. Definicja portu do którego podłączona jest dioda
    2. ustawienie wybranego portu jako wyjście cyfrowe - metoda *setup()*
    3. zapis wartości *HIGH / LOW* do wyjścia cyfrowego
  1. **Przykładowe rozwiązanie:** projekt ***00\_blink***.

1. Zapis do portu szeregowego
   1. **Cel zadania:** Zadanie polega na poznaniu portu szeregowego - możliwość wyświetlenia informacji. W zadaniu należy wyświetlić zmienną zawierającą informację o ilości przejścia pętli *loop().*
   2. **Niezbędne kroki:**
      1. Inicjalizacja portu szeregowego na częstotliwości 9600.
      2. Definicja zmiennej określają ilość iteracji.
      3. Wyświetlenie w pętli ilości dotychczasowych iteracji.
   3. **Przykładowe rozwiązanie:** projekt ***01\_serialWrite***.
2. Odczyt z portu szeregowego
   1. **Cel zadania:** Zadanie polega na odczytaniu informacji z portu szeregowego. W zadaniu zakładam, że odczytywane będą wartości w formacie procentowym (1 - 99), które posłużą nam w kolejnych zadaniach.
   2. **Niezbędne kroki:**
      1. Inicjalizacja portu szeregowego
      2. Odczyt dwóch znaków z portu szeregowego
      3. Napisanie metody sprawdzającej format danych
      4. Wyświetlenie poprawnie odczytanych wartości
   3. **Przykładowe rozwiązanie:** projekt ***02\_serialRead***
3. PWM
   1. **Cel zadania:** Zadanie polega na zapoznaniu się z wyjściami analogowymi Arduino. Zostaną przedstawione trzy warianty ustawienia wyjścia analogowego PWM - poprzez ustawienie wartości stałego wypełnienia, poprzez ustawienie wartości wypełnienia opartej na odczycie potencjometru oraz poprzez ustawienie wartości wypełnienia na podstawie wartości wczytanej poprzez port szeregowy
   2. **Niezbędne kroki (wersja z potencjometrem):**
      1. Konstrukcja poniższego układu elektrycznego:

<Schemat elektryczny>

* + 1. Inicjalizacja wejścia i wyjścia analogowego
    2. Inicjalizacja portu szeregowego w celu wyświetlania wartości
    3. Odczyt wartości z potencjometru (1 - 1024)
    4. Napisanie metody konwertującej odczyt z potencjometru do formatu rozumianego przez PWM (0 - 255)
    5. Wyświetlenie poprawnie odczytanych wartości
  1. **Niezbędne kroki (wersja z odczytem z portu szeregowego):**
     1. Konstrukcja poniższego układu elektrycznego:

<Schemat elektryczny>

* + 1. Inicjalizacja wejścia i wyjścia analogowego
    2. Inicjalizacja portu szeregowego w celu wyświetlania wartości
    3. Odczyt wartości z portu szeregowego
    4. Napisanie metody konwertującej procentowy poziom wypełnienia do formatu rozumianego przez PWM (0 - 255)
    5. Wyświetlenie poprawnie odczytanych wartości
    6. Ustawienie wartości wypełnienia
  1. **Przykładowe rozwiązanie:** projekt ***03\_PWM***.

1. OneWire
   1. **Cel zadania:** Zadanie polega na zapoznaniu się z protokołem komunikacyjnym OneWire, czujnikiem temperatury *DS18B20*. Zadanie składa się z dwóch części: pierwsza identyfikuje adres podłączonego urządzenia, druga dokonuje pomiaru temperatury
   2. **Niezbędne kroki (identyfikacja urządzenia):**
      1. Konstrukcja poniższego układu elektrycznego:

<Schemat elektryczny>

* + 1. Import odpowiednich bibliotek: *OneWire.h* oraz *DS18B20.h*
    2. Inicjalizacja portu szeregowego w celu wyświetlania wartości
    3. Utworzenie i inicjalizacja odpowiednich zmiennych reprezentujących czujnik oraz protokół komunikacyjny OneWire
    4. Odczyt i wyświetlenie poprawnego adresu
  1. **Niezbędne kroki (wyświetlenie temperatury):**
     1. Konstrukcja poniższego układu elektrycznego:

<Schemat elektryczny>

* + 1. Import odpowiednich bibliotek: *OneWire.h* oraz *DS18B20.h*
    2. Inicjalizacja portu szeregowego w celu wyświetlania wartości
    3. Definicja tablicy odpowiadającej za odczytany poprzednio adres czujnika
    4. Utworzenie i inicjalizacja odpowiednich zmiennych reprezentujących czujnik oraz protokół komunikacyjny OneWire
    5. Wyświetlenie odczytanej temperatury na porcie szeregowym.
  1. **Przykładowe rozwiązanie:** projekt ***04\_OneWire***.