

# POLITECHNIKA POZNAŃSKA

WYDZIAŁ AUTOMATYKI, ROBOTYKI I ELEKTROTECHNIKI

INSTYTUT ROBOTYKI I INTELIGENCJI MASZYNOWEJ

ZAKŁAD STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ



## SPRAWOZDANIE

SYSTEMY MIKROPROCESOROWE (LABORATORIUM)  
[WARiE\_2021-22\_AiR\_Dz\_1\_5\_D\_LUCZAK\_21/22]

WEJŚCIA/WYJŚCIA CYFROWE (GPIO, OBSŁUGA PRZERWAŃ, LED,  
PRZYCISKI MONOSTABILNE, IMPULSATOR OBROTOWY)  
(TEMAT ZAJĘĆ)

KAROL DĘBSKI

(AUTOR I: [KAROL.DEBSKI@STUDENT.PUT.POZNAN.PL](mailto:KAROL.DEBSKI@STUDENT.PUT.POZNAN.PL))

FORMA ZAJĘĆ: LABORATORIUM

PROWADZĄCY:

DR INŻ. DOMINIK ŁUCZAK

[DOMINIK.LUCZAK@PUT.POZNAN.PL](mailto:DOMINIK.LUCZAK@PUT.POZNAN.PL)

POZNAŃ 11-10-2021 09-45

(DATA I GODZINA ZAJĘĆ)

## Spis treści

1	Zadanie #1 .....	3
1.1	Specyfikacja .....	3
1.2	Implementacja .....	3
1.3	Wyniki testów.....	3
1.4	Wnioski .....	3
2	Zadanie #2 .....	4
2.1	Specyfikacja .....	4
2.2	Implementacja .....	4
2.3	Wyniki testów.....	4
2.4	Wnioski .....	4
3	Zadanie #3_i_4.....	5
3.1	Specyfikacja .....	5
3.2	Implementacja .....	5
3.3	Wyniki testów.....	5
3.4	Wnioski .....	5
4	Zadanie #5 .....	6
4.1	Specyfikacja .....	6
4.2	Implementacja .....	6
4.3	Wyniki testów.....	6
4.4	Wnioski .....	6
5	Podsumowanie.....	7

## Zadanie #1

### 1.1 Specyfikacja

Program włącza LED, odczekuje 100 ms, zgasa ją a następnie zapala kolejną LED. Cały proces jest odtwarzany w pętli. Do weryfikacji działania programu zostanie użyte narzędzie SWV.

### 1.2 Implementacja

*Listing 1 Główna pętla programu*

```
while (1)
{
    HAL_GPIO_WritePin(LD3_GPIO_Port,LD3_Pin,GPIO_PIN_SET);
    LD3_state=HAL_GPIO_ReadPin(LD3_GPIO_Port, LD3_Pin);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LD3_GPIO_Port,LD3_Pin,GPIO_PIN_RESET);
    LD3_state=HAL_GPIO_ReadPin(LD3_GPIO_Port, LD3_Pin);
    HAL_GPIO_WritePin(LD4_GPIO_Port,LD4_Pin,GPIO_PIN_SET);
    LD4_state=HAL_GPIO_ReadPin(LD4_GPIO_Port, LD4_Pin);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LD4_GPIO_Port,LD4_Pin,GPIO_PIN_RESET);
    LD4_state=HAL_GPIO_ReadPin(LD4_GPIO_Port, LD4_Pin);
    HAL_GPIO_WritePin(LD6_GPIO_Port,LD6_Pin,GPIO_PIN_SET);
    LD6_state=HAL_GPIO_ReadPin(LD6_GPIO_Port, LD6_Pin);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LD6_GPIO_Port,LD6_Pin,GPIO_PIN_RESET);
    LD6_state=HAL_GPIO_ReadPin(LD6_GPIO_Port, LD6_Pin);
    HAL_GPIO_WritePin(LD5_GPIO_Port,LD5_Pin,GPIO_PIN_SET);
    LD5_state=HAL_GPIO_ReadPin(LD5_GPIO_Port, LD5_Pin);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LD5_GPIO_Port,LD5_Pin,GPIO_PIN_RESET);
    LD5_state=HAL_GPIO_ReadPin(LD5_GPIO_Port, LD5_Pin);
    /* USER CODE END WHILE */
    MX_USB_HOST_Process();

    /* USER CODE BEGIN 3 */
}
```

Najpierw zostaje włączona dioda LD3 poprzez wygenerowanie stanu wysokiego na pinie z nią połączonym. Potem następuje oczekiwanie 100 ms, dioda jest wyłączana poprzez wygenerowanie stanu niskiego na pinie z nią połączonym. Kolejno zapala się kolejna dioda i proces się powtarza.

### 1.3 Wynik testów



Rys. 1 Podgląd SWV dla stanu 4 diod

### 1.4 Wnioski

Podgląd SWV wykazał że wartości stanu diod się zmieniają co oznacza że w istocie diody są włączane/wyłączane w pętli. Do weryfikacji działania programu zostanie użyte narzędzie SWV.

## Zadanie #2

### 1.1 Specyfikacja

Program przy naciśnięciu przycisku zmienia obecny stan diody na przeciwny.

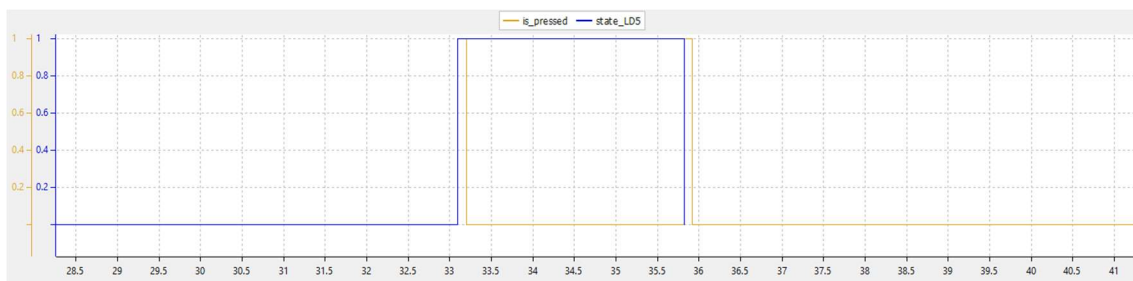
### 1.2 Implementacja

*Listing 2 Główna pętla programu*

```
while (1)
{
    state_swith1=HAL_GPIO_ReadPin(User_button_GPIO_Port,User_button_Pin);
    HAL_Delay(100);
    state_swith2=HAL_GPIO_ReadPin(User_button_GPIO_Port,User_button_Pin);
    is_pressed=state_swith1==1 && state_swith2==0;
    if(is_pressed){
        state_LD5=HAL_GPIO_ReadPin(LD5_GPIO_Port,LD5_Pin);
        if(!state_LD5){
            HAL_GPIO_WritePin(LD5_GPIO_Port,LD5_Pin,GPIO_PIN_SET);
        }else{
            HAL_GPIO_WritePin(LD5_GPIO_Port,LD5_Pin,GPIO_PIN_RESET);
        }
    }
    /* USER CODE END WHILE */
    MX_USB_HOST_Process();
    /* USER CODE BEGIN 3 */
}
```

By wykryć zbocze narastające na przycisku program najpierw sprawdza stan przycisku i go zapamiętuje. Następnie odczekiwane jest 100 ms by użytkownik zdążył podnieść palec z przycisku. W kolejnym kroku sprawdzany jest ponownie stan. Jeśli przycisk był wciśnięty to zmienia się stan diody na przeciwny do poprzedniego.

### 1.3 Wynik testów



Rys. 2 Podgląd SWV dla stanu przycisku a także dla stanu LED

### 1.4 Wnioski

Podgląd SWV wykazał że impuls 'is\_pressed' zmienia stan diody na przeciwny.

## Zadanie #3\_i\_4

### 1.1 Specyfikacja

Program zmienia stan diody na przeciwny do obecnego z wykorzystaniem przerwania przypisanego do przycisku. Do weryfikacji działania programu zostanie użyte narzędzie SWV.

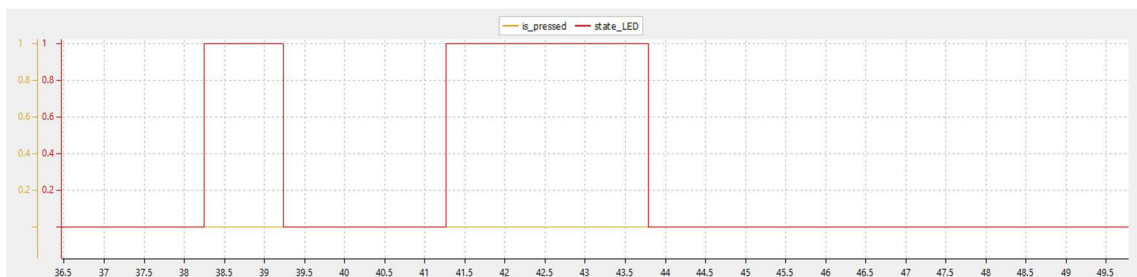
### 1.2 Implementacja

*Listing 3 Funkcja przerwania*

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == Button_Pin){
        HAL_GPIO_TogglePin(LD5_GPIO_Port, LD5_Pin);
        is_pressed=1;
        state_LED=HAL_GPIO_ReadPin(LD5_GPIO_Port, LD5_Pin);
    }
    is_pressed=0;
}
```

Po wykryciu zbocza narastającego sprawdzane jest to czy to dotyczy pinu „Button”. Jeśli tak to zmieniany jest stan LED na przeciwny od obecnego.

### 1.3 Wynik testów



Rys. 3 Podgląd stanu wciśnięcia przycisku i stanu LED

### 1.4 Wnioski

Podgląd SWV wykazał że LED zaczyna świecić gdy przycisk jest wciśnięty.

## Zadanie #5

### 1. Specyfikacja

Program zapala LED gdy pierwszy przycisk zostanie wciśnięty. Gdy drugi przycisk zostanie wciśnięty gasi LED. Do weryfikacji działania programu zostanie użyte narzędzie SWV.

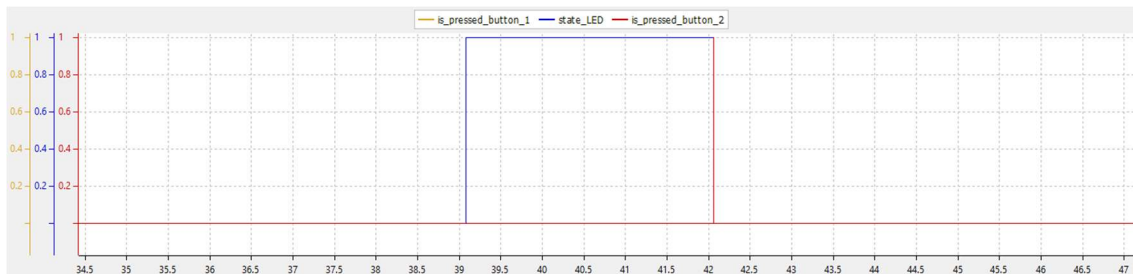
### 2. Implementacja

*Listing 4 Funkcja przerwania*

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == Button_1_Pin){
        HAL_GPIO_WritePin(LD5_GPIO_Port, LD5_Pin,GPIO_PIN_SET);
        is_pressed_button_1=1;
        state_LED=HAL_GPIO_ReadPin(LD5_GPIO_Port, LD5_Pin);
        is_pressed_button_1=0;
    }
    if(GPIO_Pin == Button_2_Pin){
        HAL_GPIO_WritePin(LD5_GPIO_Port, LD5_Pin,GPIO_PIN_RESET);
        is_pressed_button_2=1;
        state_LED=HAL_GPIO_ReadPin(LD5_GPIO_Port, LD5_Pin);
        is_pressed_button_2=0;
    }
}
```

Po wykryciu zbocza opadającego aktywowane jest przerwanie. Jeśli pierwszy przycisk został wciśnięty to LED się zapala. Odczytywany jest stan diody do wykresu w SWV a także zapisywana jest informacja o wciśnięciu przycisku pierwszego. Gdy przycisk drugi został wciśnięty to LED gaśnie i ponownie jest sprawdzany stan LED do wykresu SWV a także zapisywana jest informacja o wciśnięciu przycisku drugiego.

### 3. Wynik testów



*Rys. 4 Podgląd stanu przycisku pierwszego i drugiego a także stanu LED*

### 4. Wnioski

Podgląd SWV wykazał że LED się zapala gdy przycisk pierwszy jest wciśnięty a gaśnie gdy przycisk drugi jest wciśnięty.

## Podsumowanie

Zadanie 3 i 4 zostało połączone ze sobą ze względu na to że początek zadania 4 jest zadaniem 3. Nie zostało ukończone zadanie 5 podpunkt d, ponieważ nie do końca rozumiem pojęcia konfiguracji sprzętowej. Zadanie 6 nie zostało ukończone ze względu na to, że nie posiadam impulsatora obrotowego. Wykresy zmiennych w SWV, w zadaniu 3, 4 i 5, nie są jednoznaczne – linie nachodzą na siebie.