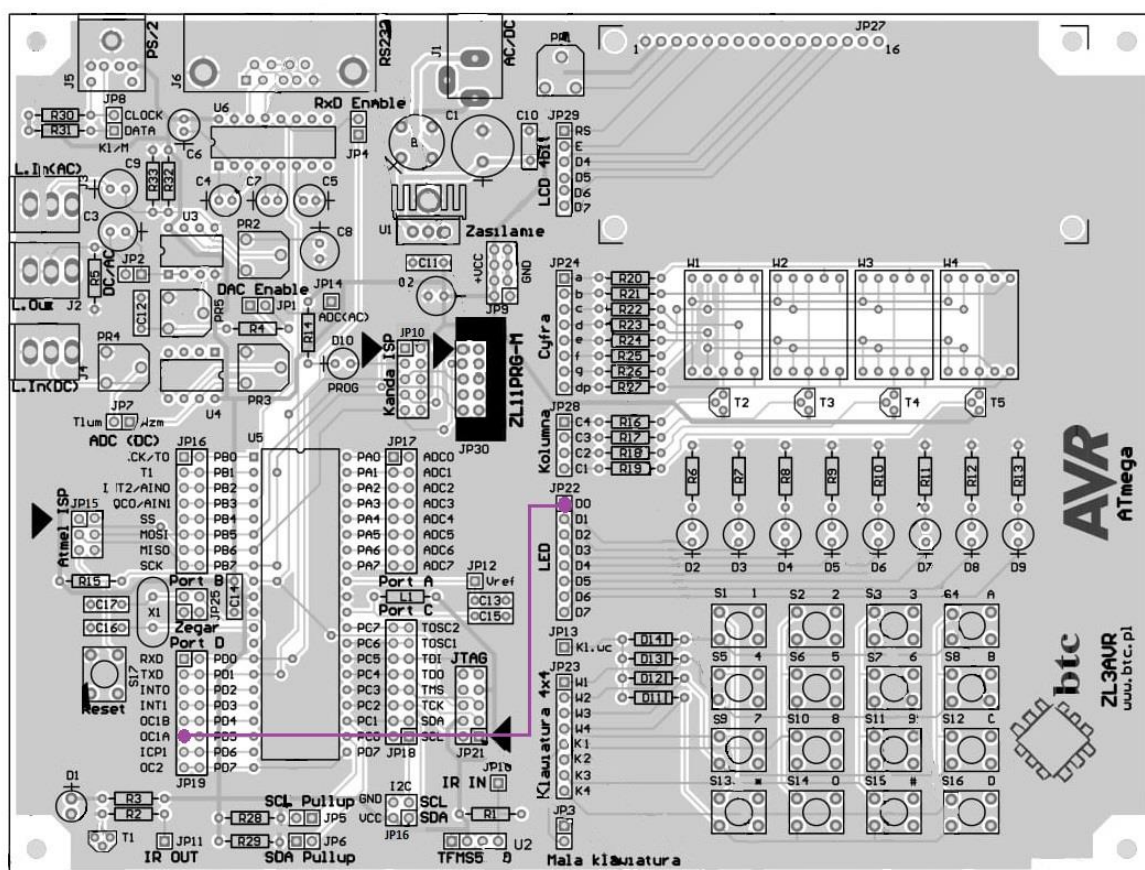
	<b>Aleksandra Kondraciuk</b> <b>grupa 5.1.1</b>	<b>TYTUŁ ĆWICZENIA</b>	<b>NUMER ĆWICZENIA</b>	<b>DATA</b>
<b>LABORATORIUM TECHNIKI MIKROPROCESOROWEJ</b>		<b>ZASTOSOWANIE TIMERÓW</b>	<b>5</b>	<b>21.11.2018r.</b>

## POPRAWA ZADANIA 3

### Treść zadania:

Korzystając z przerwań i TIMERA 1, napisz program impulsujący świeceniem LED. Czas świecenia 500ms, czas zgaszenia 300ms. Zastosuj tryb PWM. (C)

a) Schemat elektryczny zestawu ZL3AVR na którym widoczne jest zastosowane w zadaniu połączenie.



### Kod źródłowy:

```
#define F_CPU 1000000L      //ustawienie częstotliwości taktowania mikrokontrolera
#include <avr\io.h>          //dołączenie standardowej biblioteki
#include <avr\interrupt.h>   //dołączenie biblioteki

//procedura obsługi przerwania wywołanego zgodnym porównaniem
ISR(TIMER1_COMPA_vect) {

    PORTD^=(1<<5);    //zmiana stanu diody na linii 5 portu D
}

//procedura obsługi przerwania wywołanego przepełnieniem licznika
ISR(TIMER1_OVF_vect) {

    PORTD^=(1<<5);    //zmiana stanu diody na linii 5 portu D
}

int main(void) {

    DDRD=255;          //ustawienie portu D na wyjście

    TCCR1B|=(1<<WGM13) | (1<<WGM12);    //ustawienie trybu Fast PWM dla timer1
    TCCR1A|=(1<<WGM11);

    ICR1=800;          //wartość, po której nastąpi wyzerowanie licznika
    OCR1A=300;         //ustawienie wartości (300ms) dla OCR1A, po której ma nastąpić
                        //wywołanie obsługi przerwania - tyle czasu dioda będzie zgaszona

    TIMSK|=(1<<OCIE1A);    //uaktywnienie przerwania generowanego zgodnym porównaniem
    TIMSK|=(1<<TOIE1);     //uaktywnienie przerwania generowanego przepełnieniem

    TCCR1B|=(1<<CS12) | (1<<CS10);    //prekaler 1024

    sei();              //globalne zezwolenie na przerwanie

    while(1) {
        //pętla nieskończona
    }
}
```

### Wnioski:

Ustawienie bitów WGM13, WGM12 w rejestrze TCCR1B pozwoliło na skonfigurowanie TIMERA1 do pracy w trybie Fast PWM. Dzięki ustawieniu bitów CS12 i CS10 częstotliwość preskalera wynosi 1:1024.

Aby poprawić poprzednie błędy w działaniu programu użyłam drugiego przerwania, generowanego przepełnieniem licznika, aby po jego wyzerowaniu następowała zmiana stanu na diodzie.

Zatem program działa tak, że licznik liczy od 0 do wartości w OCR1A (300) i następuje przerwanie generowane zgodnym porównaniem. W obsłudze przerwania zapalana jest dioda. Licznik liczy dalej (od 300) do wartości w ICR1 (800) i następuje przepełnienie, generujące przerwanie, w którym dioda jest gaszona. Dzięki temu dioda jest zgaszona przez 300ms i zapalona przez 500ms.