**Podstawy techniki mikroprocesorowej**

**Laboratorium 1.1**

Autorzy:

Weronika Jakubowska - 248931

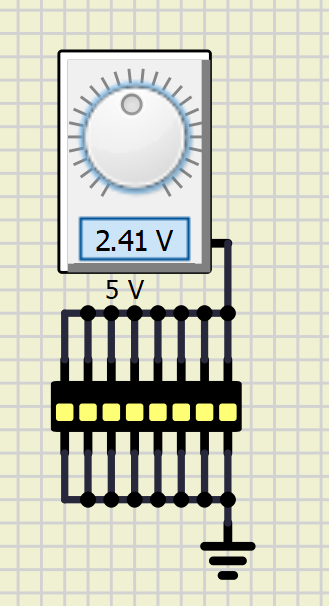
Rafał Rzewucki – 248926

Prowadzący:

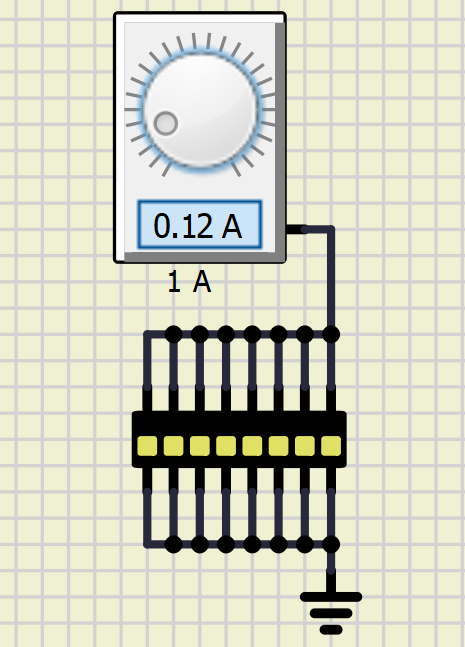
mgr inż. Wojciech Tarnawski

# Uruchamianie wszystkich diod led

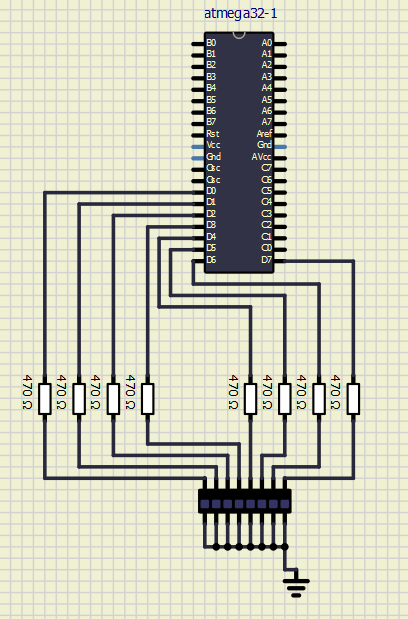
1. Sposób pierwszy



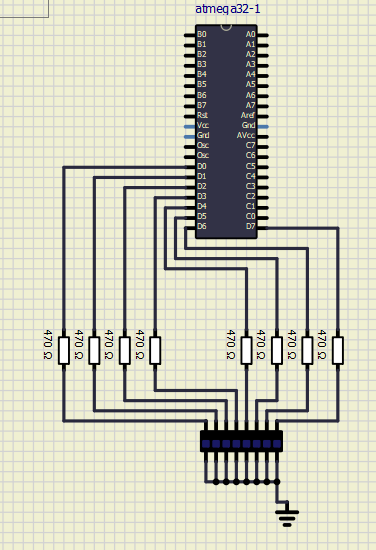
1. Sposób drugi



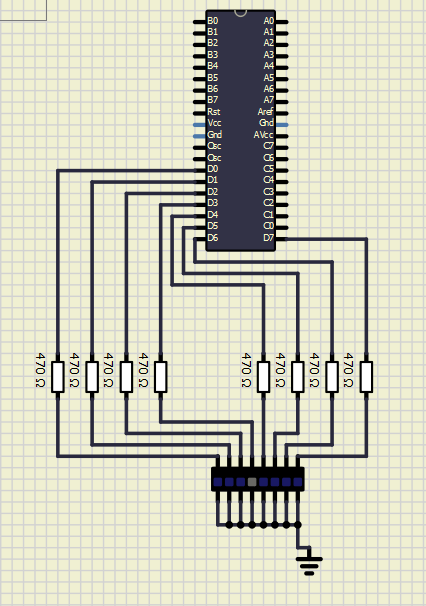
1. Sposób trzeci



II)



III)



# KOD na mruganie wszystkich diod

**#include** <avr/io.h>

**#include** <stdlib.h>

**#include** <string.h>

**#include** <util/delay.h>

**#ifndef** \_BV

**#define** \_BV(bit) (1<<(bit))

**#endif**

**#ifndef** sbi // 1

**#define** sbi(reg,bit) reg |= (\_BV(bit))

**#endif**

**#ifndef** cbi // 0

**#define** cbi(reg,bit) reg &= ~(\_BV(bit))

**#endif**

**#ifndef** tbi // 0

**#define** tbi(reg,bit) reg ^= (\_BV(bit))

**#endif**

**#define** DELAY 500

// DDRD Data Direction dla portu D

**int** **main**() {

**for**(**int** i =0; i< 8; i++) // pętla po portach D 0-7

sbi(DDRD, i); // ustawianie portu jako wyjście.

**while** (1) { // pętla gwarantująca zapętlanie się programu

**for**(**int** i =0; i< 8; i++)

sbi(PORTD, i); // ustawianie stanu portu na 1

**\_delay\_ms**(DELAY); // Czekanie

**for**(**int** i =0; i< 8; i++)

cbi(PORTD, i); // ustawianie stanu portu na 0

**\_delay\_ms**(DELAY); // Czekanie

**\_delay\_ms**(DELAY); // Czekanie

}

**return** 0;

}

# KOD na biegający punkt:

**#include** <avr/io.h>

**#include** <stdlib.h>

**#include** <string.h>

**#include** <util/delay.h>

**#ifndef** \_BV

**#define** \_BV(bit) (1<<(bit))

**#endif**

**#ifndef** sbi // 1

**#define** sbi(reg,bit) reg |= (\_BV(bit))

**#endif**

**#ifndef** cbi // 0

**#define** cbi(reg,bit) reg &= ~(\_BV(bit))

**#endif**

**#define** DELAY 500

// DDRD Data Direction dla portu D

**int** **main**() {

**for**(**int** i =0; i< 8; i++) // pętla po portach D 0-7

sbi(DDRD, i); // ustawianie portu jako wyjście.

**while** (1) { // pętla gwarantująca zapętlanie się programu

**for**(**int** i =0; i< 8; i++){ // pętla po wybranych portach

sbi(PORTD, i); // ustawianie stanu portu na 1

**\_delay\_ms**(DELAY); // Czekanie

cbi(PORTD, i); // ustawianie stanu portu na 0

**\_delay\_ms**(DELAY); // Czekanie

}

}

**return** 0;

}