

```

#include <avr/io.h>

#define test_delay 500
#define delay_pulse 500
#define pin_53 PB0 // rood
#define pin_51 PB2 // groen
#define pin_50 PB3 // geel links
#define pin_48 PL1 // geel rechts

#define rood pin_53
#define groen pin_51
#define geel_links pin_50
#define geel_rechts pin_48

#include <util/delay.h>

void doorvaartVerbieden(){

    /*!
    * Deze functie zet de rode lichten aan
    *
    * Dit geeft aan dat
    */
    PORTB &= ~(1 << groen);
    PORTB &= ~(1 << geel_links);
    PORTB &= ~(1 << geel_rechts);
    PORTB |= (1 << rood);
}

void aanstondsToestaan(){
    PORTB &= ~(1 << geel_links);
    PORTB &= ~(1 << geel_rechts);
    PORTB |= (1 << rood);
    PORTB |= (1 << groen);
}

void doorvaartToestaan(){

    /*
    */
    PORTB &= ~(1 << rood);
    PORTB &= ~(1 << geel_rechts);
    PORTB &= ~(1 << geel_links);
    PORTB |= (1 << groen);
}

void doorvaartToestaanUitzondering(){

    /*

```

```

*/

PORTB &= ~(1 << rood);
PORTB &= ~(1 << geel_rechts);
PORTB &= ~(1 << geel_links);
PORTB |= (1 << rood);
for(int i = 0; (delay_pulse * i) < test_delay; i++){
    PORTB ^= (1 << groen);
    _delay_ms(delay_pulse);
}

}

void doorVaartverbiedenBrugOpen(){
    /*!
    *Deze functie zet de lichten op rood bij een open brug
    *
    *Dit geeft aan dat ondanks dat de brug open is je hier niet doorheen mag varen
    */
    doorvaartVerbieden();
}

void doorvaartToestaanGeslotenBrugTweerichting(){
    /*!
    *Deze functie zorgt ervoor dat de rode lichten aangaan
    *aan beide kanten en één van de gele lichten aangaat.
    *Dit geeft aan dat er verkeer doorheen mag ookal is de brug gesloten,
    *daarnaast geeft dit aan dat er verkeer vanaf de andere kant kan komen.
    */
    doorvaartVerbieden();
    PORTB |= (1 << geel_links);
}

void doorvaartToestaanGeslotenBrugEenrichting(){
    /**
    *Deze functie zorgt ervoor dat de rode lichten aangaan
    *aan beide kanten en één van de gele lichten aangaat.
    *Dit geeft aan dat er verkeer doorheen mag ookal is de brug gesloten,
    *daarnaast geeft dit aan dat er geen verkeer vanaf de andere kant komt.
    */
    doorvaartVerbieden();
    PORTB |= (1 << geel_links);
    PORTB |= (1 << geel_rechts);
}

int main(void) {
    // geel rechts
    DDRB |= (1 << geel_rechts);
    // geel links
    DDRB |= (1 << geel_links);
    // groen
    DDRB |= (1 << groen);
    // rood
    DDRB |= (1 << rood);
    while (1) {
        doorvaartVerbieden();

```

```
        _delay_ms(test_delay);  
        aanstondsToestaan();  
        _delay_ms(test_delay);  
        doorvaartToestaanUitzondering();  
        _delay_ms(test_delay);  
        doorvaartToestaan();  
        _delay_ms(test_delay);  
        doorvaartToestaanGeslotenBrugTweerichting();  
        _delay_ms(test_delay);  
        doorvaartToestaanGeslotenBrugEenrichting();  
        _delay_ms(test_delay);  
    }  
}
```