/\*  
 \* Opdracht 4.3 - Led There Be Light  
 \*   
 \* Eigenaar:  
 \* Maarten Paauw <s1094220@student.hsleiden.nl>  
 \* s1094220  
 \* INF3C  
 \*   
 \* Versie: 1  
 \* Aangemaakt: 12 maart 2018  
 \* Gewijzigd: 27 maart 2018  
 \*/  
  
// Baud rate.  
#define BAUD 9600  
  
// LED pinnen.  
#define LED\_RED PB1  
#define LED\_GREEN PB2  
#define LED\_BLUE PB3  
  
// Includes.  
#include <avr/io.h>  
#include <util/setbaud.h>  
#include <avr/interrupt.h>  
  
// Count.  
int count = 0;  
  
// Variabele voor de kleur code (standaard rood).  
char color = 'r';  
  
// Uit het bestand USART.  
void initUSART(void)  
{  
 UBRR0H = UBRRH\_VALUE;  
 UBRR0L = UBRRL\_VALUE;  
  
 #if USE\_2X  
 UCSR0A |= (1 << U2X0);  
 #else  
 UCSR0A &= ~(1 << U2X0);  
 #endif  
  
 // RXCIE0 zorgt ervoor dat de interrupt triggert op character.  
 UCSR0B = (1 << RXCIE0) | (1 << TXEN0) | (1 << RXEN0);  
 UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);  
}  
  
// Uit het bestand USART.  
uint8\_t receiveByte(void)  
{  
 loop\_until\_bit\_is\_set(UCSR0A, RXC0);  
 return UDR0;  
}  
  
// Uit het bestand USART.  
void transmitByte(uint8\_t data)  
{  
 loop\_until\_bit\_is\_set(UCSR0A, UDRE0);  
 UDR0 = data;  
}  
  
// Uit het bestand USART.  
void printString(const char myString[])  
{  
 uint8\_t i = 0;  
 while (myString[i])  
 {  
 transmitByte(myString[i]);  
 i++;  
 }  
}  
  
// Overflow timer.  
void initTimerOverflow()  
{  
 // Timer mask.  
 TIMSK0 |= (1 << TOIE0);  
  
 // Timer instellingen.  
 TCCR0B |= (1 << CS02) | (1 << CS00);  
}  
  
// LED timers.  
void initPWMTimerLED()  
{  
 // Timer 1 (Voor rood en groen).  
 TCCR1A |= (1 << WGM10) | (1 << WGM12); // Fast PWM.  
 TCCR1B |= (1 << CS11); // PWM freq.  
  
 // Timer 2 (Voor blauw).  
 TCCR2A |= (1 << WGM20) | (1 << WGM21); // Fast PWM.  
 TCCR2B |= (1 << CS21); // PWM freq.  
  
 // Rode en groene led.  
 TCCR1A |= (1 << COM1A1) | (1 << COM1B1);  
  
 // Blauwe led.  
 TCCR2A |= (1 << COM2A1);  
}  
  
// Verander de led van kleur.  
void changeColor(int r, int g, int b) {  
  
 // Verander de rode led.  
 OCR1A = ~r;  
  
 // Verander de groene led.  
 OCR1B = ~g;  
  
 // Verander de blauwe led.  
 OCR2A = ~b;  
}  
  
// Zet de kleur en stuur een bericht  
void setColorAndMessage(char c, char message[]) {  
   
 // laat de gebruiker weten welke kleur het wordt.  
 printString(message);  
  
 // Reset de count.  
 count = 0;  
  
 // Sla het op in een variabele.  
 color = c;  
}  
  
// Timer overflow interrupt.  
ISR (TIMER0\_OVF\_vect)  
{  
 // Tel de count op, voor de brightness.  
 count++;  
  
 // Switch om te kijken om welke kleur het gaat.  
 switch(color)  
 {  
  
 // De kleur rood.  
 case 'r':  
  
 // Verander de led kleur.  
 changeColor(count, 0, 0);  
  
 // Break de case.  
 break;  
   
 // De kleur groen.  
 case 'g':  
  
 // Verander de led kleur.  
 changeColor(0, count, 0);  
  
 // Break de case.  
 break;  
  
 // De kleur blauw.  
 case 'b':  
  
 // Verander de led kleur.  
 changeColor(0, 0, count);  
  
 // Break de case.  
 break;  
 }  
}  
  
// Als er geschreven wordt naar de seriele verbinding.  
ISR (USART\_RX\_vect)  
{  
 // Kijk welke kleur het is.  
 switch (UDR0)  
 {  
  
 // Als het rood is.  
 case 'r':  
  
 // Zet de kleur en laat de gebruiker weten dat de kleur rood wordt.  
 setColorAndMessage('r', "De kleur wordt rood.  
");  
  
 // Break de case.  
 break;  
   
 // Als het groen is.  
 case 'g':  
  
 // Zet de kleur en laat de gebruiker weten dat de kleur groen wordt.  
 setColorAndMessage('g', "De kleur wordt groen.  
");  
  
 // Break de case.  
 break;  
  
 // Als het blauw is.  
 case 'b':  
  
 // Zet de kleur en laat de gebruiker weten dat de kleur blauw wordt.  
 setColorAndMessage('b', "De kleur wordt blauw.  
");  
  
 // Break de case.  
 break;  
  
 // De standaard switch case.  
 default:  
  
 // Meld aan de gebruiker dat er een toets ingedrukt moet worden.  
 printString("Toets R, G of B in.  
");  
  
 // Break de case.  
 break;  
 }  
}  
  
// Main methode.  
int main(void)  
{  
 // Initialiseer de RGB LED pinnen.  
 DDRB = (1 << LED\_RED) | (1 << LED\_GREEN) | (1 << LED\_BLUE);  
  
 // Initialiseer de USART.  
 initUSART();  
  
 // Initialiseer de overflow timer.  
 initTimerOverflow();  
  
 // Initialiseer de led timer.  
 initPWMTimerLED();  
  
 // Zet de interrupts aan.  
 sei();  
  
 // Loop voor altijd.  
 while (1) { }  
  
 // Geef niets terug.  
 return 0;  
}