

```

1  /*
2   * main.c
3   *
4   * Created: 03/03/2023 10:23:33 PM
5   * Author: Kirollos
6   */
7  #define BAUD_RATE 9600
8  #define F_CPU 1000000
9  #include <avr/io.h>
10 #include <avr/interrupt.h>
11 #include <util/delay.h>
12 #include "UART.h"
13
14 #define MOTOR_INC 'I'
15 #define MOTOR_DEC 'D'
16 #define MOTOR_FWD 'F'
17 #define MOTOR_REV 'R'
18 #define MOTOR_STOP 'S'
19
20 #define LED_ON 'o'
21 #define LED_OFF 'f'
22
23 const float stepsize = 0.1;
24 int main(void) {
25     float duty_cycle = 0;
26     TCCR0=0b01110101; // Configure TCCR0 as explained in the article
27     OCR0=255; // Set OCR0 to 255 for initial duty cycle = 0 and the motor is not
        rotating
28     DDRB |= 1<<3;
29     DDRA = 1;
30     DDRD |= 1<<2 | 1<<3;
31     PORTA |= 1;
32     static volatile uint8_t cmd_buffer[1];
33     /* Init UART driver. */
34     UART_cfg my_uart_cfg;
35     /* Set USART mode. */
36     my_uart_cfg.UBRRH_cfg = (BAUD_RATE_VALUE)&0xFF;
37     my_uart_cfg.UBRRH_cfg = (((BAUD_RATE_VALUE)&0xFF00)>>8);
38     my_uart_cfg.UCSRA_cfg = 0;
39     my_uart_cfg.UCSRB_cfg = (1<<RXEN) | (1<<TXEN) | (1<<TXCIE) | (1<<RXCIE);
40     my_uart_cfg.UCSRC_cfg = (1<<URSEL) | (3<<UCSZ0);
41     UART_Init(&my_uart_cfg);
42     sei();
43     while(1) {
44         /* Receive the full buffer command. */
45         UART_ReceivePayload(cmd_buffer, 1);
46         /* Poll until reception is complete. */
47         while(0 == UART_IsRxComplete());
48         /* Parse command buffer. */
49         switch(cmd_buffer[0]) {
50             case LED_ON: {
51                 PORTA |= 1;

```

```
52         break;
53     }
54     case LED_OFF: {
55         PORTA &= ~1;
56         break;
57     }
58     case MOTOR_INC: {
59         if((duty_cycle+stepsize) < 1.0) {
60             duty_cycle += stepsize;
61         }
62         else
63             duty_cycle = 1;
64         break;
65     }
66     case MOTOR_DEC: {
67         if((duty_cycle-stepsize) > 0) {
68             duty_cycle -= stepsize;
69         }
70         else
71             duty_cycle = 0;
72         break;
73     }
74     case MOTOR_FWD: {
75         PORTD |= 1<<2;
76         PORTD &= ~(1<<3);
77         break;
78     }
79     case MOTOR_REV: {
80         PORTD |= 1<<3;
81         PORTD &= ~(1<<2);
82         break;
83     }
84     case MOTOR_STOP: {
85         PORTD &= ~(1<<2);
86         PORTD &= ~(1<<3);
87         break;
88     }
89     default: {
90         /* Do nothing. */
91     }
92 }
93 OCR0 = 255 - 255*duty_cycle; // d = (255-OCR0)/255 // 255 - d*255
94 _delay_ms(250);
95 }
96 return 0;
97 }
```