```
/* CPU freqvensce */
#define F_CPU 8000000UL
 /* Libraries*/
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

/* definition of static variables*/
#define  Trigger_pin    PORTA0         /* Trigger pin */

/* global variables */
long TimerOverflow = 0;
char string[10];
long count;
double distance;
uint8_t right1=0;
uint8_t left1 =0;
uint8_t  turn = 0;
uint8_t  lastturn = 0;

/* functions definition*/
void move(void);
double measureDistance(void);
void OC3ACounter(int speed);
void OC3BCounter(int speed);
void SetSpeedOC3A(int speed);
void SetSpeedOC3B(int speed);
void stop();


/* functions main one*/
int main(void)
{

        OC3ACounter(400); // set the counter Timers
        OC3BCounter(500); // set the counter timers

        DDRA |= (1 << PORTA0) | (1 << PORTA1) | (1 << PORTA2);                  /* Make trigger
pin as output */
        PORTD |= (1 << PORTD6);             /* Turn on Pull-up */
        //DDRB &= ~(1 << PORTB7); // right engine turning left
        //DDRB &= ~(1 << PORTB6); // left engine  turning right
```

```
	sei();			/* Enable global interrupt */
	TIMSK1 = (1 << TOIE1);	/* Enable Timer1 overflow interrupts */
	TCCR1A = 0;		/* Set all bit to zero Normal operation */




	while(1)
	{
		distance = measureDistance();  /* distance from ultrasonic sensor */
		if (distance < 15) /* stop the motor if the object is nearby */
		{
			PORTA |= (1 << PORTA2); // red light
			DDRB &= ~(1 << PORTB7); // right engine turning left
			DDRB &= ~(1 << PORTB6); // left engine  turning right

			PORTA &= ~(1 << PORTA1); // turn of blue light
			_delay_ms(500);
		}

		for (int x = 0; x < 29000; x++) /* for loop to delay the if any ultra sound jumps
back about and for loop delay is about 300ms */
		{
			if(distance >= 15 ) /* if object is in nearby the 20cm move the move the
car forward */
			{
				move();
			}
			else{ /* otherise break the loop to not delay */
				break;
			}
		}



	}
}

/* functions for  measure distance by ultrasonic sensors*/
double measureDistance(void){
	/* Give 10us trigger pulse on trig. pin to HC-SR04 */
	PORTA |= (1 << Trigger_pin);
```

```
	_delay_us(10);
	PORTA &= (~(1 << Trigger_pin));

	TCNT1 = 0;	/* Clear Timer counter */
	TCCR1B = 0x41;	/* Capture on rising edge, No prescaler*/
	TIFR1 = 1<<ICF1;	/* Clear ICP flag (Input Capture flag) */
	TIFR1 = 1<<TOV1;	/* Clear Timer Overflow flag */

	/*Calculate width of Echo by Input Capture (ICP) */

	while ((TIFR1 & (1 << ICF1)) == 0);/* Wait for rising edge */
	TCNT1 = 0;	/* Clear Timer counter */
	TCCR1B = 0x01;	/* Capture on falling edge, No prescaler */
	TIFR1 = 1<<ICF1;	/* Clear ICP flag (Input Capture flag) */
	TIFR1 = 1<<TOV1;	/* Clear Timer Overflow flag */
	TimerOverflow = 0;/* Clear Timer overflow count */

	while ((TIFR1 & (1 << ICF1)) == 0);/* Wait for falling edge */
	count = ICR1 + (65535 * TimerOverflow);	/* Take count */
	/* 8MHz Timer freq, sound speed =343 m/s */
	return (double)count / 466.47;
}

/* functions for move the motors to follows the lines */
void move(void)
{
	PORTA &= ~(1 << PORTA2);
	PORTA |= (1 << PORTA1);
	if (( ( (PIND&(1 << PORTD5)) >> PORTD5) == 1 ) & (( (PIND&(1 << PORTD3)) >>
PORTD3) == 1 ) & (( (PIND&(1 << PORTD4)) >> PORTD4) == 1))
	{
		DDRB &= ~(1 << PORTB7); // right engine turning left
		DDRB &= ~(1 << PORTB6); // left engine  turning right

		SetSpeedOC3A(390);
		SetSpeedOC3B(490);

		int rightTurn = 0;
		int right1 = 0;
		while ( (( (PIND&(1 << PORTD3)) >> PORTD3) == 1 ) & (turn == 0)) // höger
		{
			DDRB = 0b01000000;
			rightTurn = 1;
```

```
                right1 = 1;

        }


        int leftTurn = 0;
        int left1 = 0;
        while ((( (PIND&(1 << PORTD5)) >>PORTD5) == 1) & (turn == 1)) // vänster
        {
                DDRB = 0b10000000;
                left1 = 1;
                lastturn = 1;

        }


        turn++;

        if (turn > 1 & lastturn ==1)
        {
                turn = 0;
        }


    }else if (( ( (PIND&(1 << PORTD5)) >> PORTD5) == 0 ) & (( (PIND&(1 << PORTD3)) >>
PORTD3) == 0 ) & (( (PIND&(1 << PORTD4)) >> PORTD4) == 0))
    {
        if (right1)
        {
                while (( ( (PIND&(1 << PORTD5)) >> PORTD5) == 0 ) & (( (PIND&(1 <<
PORTD3)) >> PORTD3) == 0 ) & (( (PIND&(1 << PORTD4)) >> PORTD4) == 0))
                {
                        DDRB = 0b01000000;
                }
        }else if (left1)
        {
                while (( ( (PIND&(1 << PORTD5)) >> PORTD5) == 0 ) & (( (PIND&(1 <<
PORTD3)) >> PORTD3) == 0 ) & (( (PIND&(1 << PORTD4)) >> PORTD4) == 0))
                {
                        DDRB = 0b10000000;
                }
        }
```

```
                if (!left1 & !right1)
                {
                        DDRB = 0b00000000;
                        PORTA &= ~(1 << PORTA1);// lamp blue turn off
                }
                right1 =0;
                left1  =0;

        }
        else{

                PORTA &= ~(1 << PORTA1);// lamp blue turn off
                PORTA |= (1 << PORTA1);  // turn on blue lamp
                SetSpeedOC3A(490);
                SetSpeedOC3B(590);
                // hålla sig i linje
                if( ( ( (PIND&(1 << PORTD5)) >> PORTD5) == 1 ) & (( (PIND&(1 << PORTD3))
>> PORTD3) == 0 ) & (( (PIND&(1 << PORTD4)) >> PORTD4) == 0) ){ // gå fram
                        DDRB = 0b11000000;
                        right1=0;
                        left1 =0;
                }
                if(( (PIND&(1 << PORTD3)) >> PORTD3) == 1 ){ // höger
                        DDRB = 0b01000000;
                        right1 = 1;
                }
                if(( (PIND&(1 << PORTD4)) >> PORTD4) == 1){ // vänster
                        DDRB = 0b10000000;
                        left1 = 1;
                }
                PORTA &= ~(1 << PORTA1);// lamp blue turn off
  }
}

/* functions for Counter timer 3A */
void OC3ACounter(int speed){


        OCR3B = 1023; // top value
        OCR3A = speed;
        TCCR3A |= (1 << COM3A1) | (1 << COM3A0) | (1<<WGM32) | (1<<WGM31) | (1 <<
WGM30);
        TCCR3B |= (1 << CS30);
```

```
}

/* functions for Counter Timer 3B */
void OC3BCounter(int speed){


        OCR3B = speed; // top value 400 to 0
        TCCR3A |= (1 << WGM33) | (1 << WGM31) | (1 << COM3B1) ;
        TCCR3B |=   (1 << CS30);
}

/* functions for interrupt */
ISR(TIMER1_OVF_vect)
{
        TimerOverflow++;       /* Increment Timer Overflow count */
}

/* functions for change the Counter Timer width */
void SetSpeedOC3A(int speed){
        OCR3A = speed;
}

/* functions for Change counter Timer pulse width*/
void SetSpeedOC3B(int speed){
        OCR3B = speed;
}

/* functions To stop all counter Timers*/
void stop(void){
        DDRB = 0x00;

        OCR0A, OCR0B, OCR3B, OCR3A = 0;


        TCCR0A ,TCCR0B ,TCCR3A ,TCCR3B = 0;
}
```