```
//made by darsh
// line follower robot. sensor 1 has threshold 75 and sensor 2 has 25 as
suited
#include<avr/io.h>
#include<compat/deprecated.h>
#include<avr/delay.h>

void reverse(void); // function prototypes
void right(void);
void left(void);

void InitADC(void)
{
      ADMUX = 0;
      ADMUX |= (1 << REFS0); // Set ADC reference to AVCC
      //ADMUX |= (1 << ADLAR); // Left adjust ADC result to allow easy 8
bit reading

      ADCSRA = 0xC3;// 11000011 = prescale by 8. and  aden and adsc set

      while(!(ADCSRA & 0x10)); // conversion going on as flag is not set

      ADCSRA |= 0x10;// write flag to 1 to reset it and start a new
conversion
}


// ReadADC

unsigned int ReadADC(unsigned char channel)
{
      unsigned int volt,volt_low,volt_high;

      ADMUX = channel;

      ADCSRA = 0xC3;

      while(!(ADCSRA & 0x10));

      ADCSRA |= 0x10;
      //volt=ADCH;
      volt_low = ADCL;
      volt_high = ((unsigned int) (ADCH <<8));

      volt = volt_low | volt_high;


      return volt;
}

int main(void)
{DDRC=0xff;
DDRD=0xff;
   int x,y;
```

```
   TCCR1A =  0b10100001;     // enable 8 bit PWM, select fast non
inverting PWM


          TCCR1B = _BV(CS11) | _BV(WGM12);


   while(1)
   {
            InitADC();
            x=ReadADC(0);
            y=ReadADC(1);
            //forward();
            if(x>75 && y>25)
                  {
                        PORTC=0xff;
                        forward();
                        _delay_ms(25);
                  }
            else if(x<75 && y>25)
                  {     PORTC=0x00;
                        left();
                        _delay_ms(25);
                  }
            else if(x>75 && y<25)
                  {     PORTC=0x00;
                        right();
                        _delay_ms(25);
                  }
            else if(x<75 && y<25)
                  {     PORTC=0x00;
                        forward();
                        _delay_ms(25);
                  }
                  else
                  {_delay_ms(25);
                  }
      }

}
void forward()
{
PORTD=0b01110100;
OCR1AL = 75;
OCR1BL = 75;

}

void left()
{
PORTD= 0b01110000;
OCR1AL = 75;
OCR1BL = 75;
```

```
}

void right()
{
PORTD=0b00110100;
OCR1AL = 75;
OCR1BL = 75;



}
```