```
#define USART_BAUDRATE 9600 // This is set for serial commmuncation
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)
#include <math.h>
#include <avr/io.h>
#include <util/delay.h>

void usart_init()

{

    UCSRB |= (1 << RXEN) | (1 << TXEN);   // Turn on the transmission and
reception circuitry
    UCSRC |= (1 << URSEL) | (1<<USBS) | (1 << UCSZ0) | (1 << UCSZ1);
    // Use 8-bit character sizes

    UBRRL = BAUD_PRESCALE;
    // Load lower 8-bits of the baud rate value into the low byte of the
UBRR register
    UBRRH = (BAUD_PRESCALE >> 8); // Load upper 8-bits of the baud rate
value..
    // into the high byte of the UBRR register
}
unsigned int usart_getch()

{

    while ((UCSRA & (1 << RXC)) == 0);
    // Do nothing until data has been received and is ready to be read from
UDR
    return(UDR); // return the byte
}


void pwm_init(){
    //we would be using timers in Fast PWM mode to control the Two L293ds
    //setting all the three timers in fast pwm mode and prescaler 1
    TCCR0|=(1<<WGM01)|(1<<WGM00)|(1<<CS00)|(1<<COM01);
    TCCR2|=(1<<WGM21)|(1<<WGM20)|(1<<CS20)|(1<<COM21);
    TCCR1A|=(1<<WGM10)|(1<<COM1B1)|(1<<COM1A1);
    TCCR1B|=(1<<WGM12)|(1<<CS10);

}
int main(void)
{   uint16_t angle=0; // this would store the commands coming from the
bluetooth
    pwm_init();             // call to pwm_init to initialize the timers
    usart_init();           // call to usart_init to initialize the serial
communication
    uint8_t digit1=0,digit2=0,digit3=0;   // these three variables are used
to convert the string coming from serial communication to a number
    // Atmega32 has the pins PB3,PD4,PD5,PD7 as PWM pins  but to be able to
use them as pwm we must set these as output
    DDRB|=(1<<DDB3);    //set portB pin 3 as output
    DDRD|=(1<<DDD4)|(1<<DDD5)|(1<<DDD7);//set portd pin 4,5,7 as output
    DDRC=0b11111111;// define whole port C as output to motors

    while(1){
    if(usart_getch()=='*')          // this condition checks if an * is
received if this is true then it stores
    {                               // the next three characters in the
digit variables
```

```
    digit1=usart_getch()-48;
    digit2=usart_getch()-48;
    digit3=usart_getch()-48;
    }
    angle=100*digit1+10*digit2+digit3;// getting an angle out of string

    /* the following code segment decides how much voltage should be given
to the four motors, for this first we divide
    the whole circle into 8 equal segments of 45 degrees each. now while
moving the diagonally opposite wheels can be
considered as a single unit. we would be running the wheels such that one
set is always at full voltage and other has
lesser voltage by some factor. simple vector mathematics shows that the
ratio is tan(45 + angle). For deciding which
set has a higher voltage and the direction of the voltage we need to refer
to the corresponding segment of the circle.

For angles like 45,135, 225,315 which give an 0 of infinity in the formula
tan(45 +angle), we write separate commands.
        */
        double angle_45= M_PI*angle/180+M_PI_4;
        if(!((angle==45)|(angle==135)|(angle==225)|(angle==315)))
        {
            if((angle>=0)&&(angle<45))
            {


                OCR1AL=OCR1BL=255;
                OCR0=OCR2=fabs(255/tan(angle_45));
                PORTC=0b10101010;
            }
            else if((angle>45)&&(angle<=90))
            {

                OCR1AL=OCR1BL=255;
                OCR0=OCR2=fabs(255/tan(angle_45));
                PORTC=0b01011010;
            }
            else if((angle>=90)&&(angle<135))
            {

                OCR1AL=OCR1BL=fabs(tan(angle_45)*255);
                OCR0=OCR2=255;
                PORTC=0b01011010;
            }
            else if((angle>135)&&(angle<=180))
            {

                OCR1AL=OCR1BL=fabs(tan(angle_45)*255);
                OCR0=OCR2=255;
                PORTC=0b01010101;
            }
            else if((angle>=180)&&(angle<225))
            {

                OCR1AL=OCR1BL=255;
                OCR0=OCR2=fabs(255/tan(angle_45));
                PORTC=0b01010101;
            }
            else if((angle>225)&&(angle<=270))
            {
```

```
            OCR1AL=OCR1BL=255;
            OCR0=OCR2=fabs(255/tan(angle_45));
            PORTC=0b10100101;
        }
        else if((angle>=270)&&(angle<315))
        {

            OCR1AL=OCR1BL=fabs(tan(angle_45)*255);
            OCR0=OCR2=255;
            PORTC=0b10100101;
        }
        else if((angle>315)&&(angle<360))
        {

            OCR1AL=OCR1BL=fabs(tan(angle_45)*255);
            OCR0=OCR2=255;
            PORTC=0b10101010;

        }
        else if(angle==444)
        {
            OCR1AL=OCR1BL=0;
            OCR0=OCR2=0;}

        else if(angle==361)// this is for clockwise rotation
        {
            OCR1AL=OCR1BL=170;
            OCR0=OCR2=170;
            PORTC=0b00111001;
            //PORTC|=(1<<6);
        }
        else if(angle==362)// this is for anticlockwise rotation
        {
            OCR1AL=OCR1BL=170;
            OCR0=OCR2=170;
            PORTC=0b11000110;
        }



    }
    else if(angle==45){
        OCR1AL=OCR1BL=255;
        PORTC=0b00001010;


    }
    else if(angle==135){
        OCR0=OCR2=255;
        PORTC=0b01010000;


    }
    else if(angle==225)
    {
        OCR1AL=OCR1BL=255;
        PORTC=0b00000101;

    }
    else if(angle==315)
```

```
        {
            OCR0=OCR2=255;
            PORTC=0b10100000;
        }









    }

    }
```