

C-kod, bilen

```
#define F_CPU 8000000          //pga klockfrekvens på 8Mhz
#include <avr/io.h>
#include <inttypes.h>
#include <util/delay.h>

unsigned int direction, speed;
unsigned short int recieve;

#define CSN_LOW() PORTB &= ~ (1 << PB4);
#define CSN_HIGH() PORTB |= (1 << PB4);

void SPI_slaveInit(void) {      //SPI initiate för slaven(bilen)

    DDRB |= (1 << PB6);
    SPCR |= (1 << SPE) | (1 << SPR0); //enable SPI
}

void Drive(int direction)       //sätter på motorerna
{
    while (!(SPSR & (1 << SPIF))) //kör ända tills ny information från
master är mottagen
    {
        PORTA = direction;          //ställer in motorerna efter
önskad riktning
        _delay_ms(speed);           //beroende på fart stannar
motorerna olika länge
        PORTA = 0x12;               //stannar motorerna
        _delay_ms(speed);

    }
    if(direction == 0x12) {
        toggleLight(0b01000000); //om direktion = stanna så togglas
bakljusen av när nytt kommando ges
    }
}

void toggleLight(int light)
{
    PORTD = PORTD ^ light;          //toggla av/på lamporna
}
```

void blinkingLights(int lights)	//blinka lampor
{	
int k;	
for(k = 0; k < 7; k++)	
{	
toggleLight(lights);	
_delay_ms(500);	
k++;	
}	
}	
 void lightShow()	//ljusshow med diverse
blinkningar och snurr	
{	
int k;	
 PORTD = 0xff;	
_delay_ms(500);	
PORTD = 0x00;	
_delay_ms(50);	
PORTD = 0xff;	
_delay_ms(50);	
PORTD = 0x00;	
 for(k = 0; k < 2; k++)	
blinkingLights(0xff);	
 for(k = 0; k < 11; k++)	
{	
toggleLight(0b01000000);	
_delay_ms(50);	
toggleLight(0b01010000);	
_delay_ms(50);	
toggleLight(0b00110000);	
_delay_ms(50);	
toggleLight(0b00100100);	
_delay_ms(50);	
toggleLight(0b00001100);	
_delay_ms(50);	
toggleLight(0b00001000);	
}	
 PORTA = 0x1B;	
_delay_ms(2000);	
PORTA = 0x12;	
 toggleLight(0b01000000);	
_delay_ms(500);	
toggleLight(0b01000000);	

```

    for(k = 0; k < 5; k++)
    {
        toggleLight(0b01000000);
        _delay_ms(50);
        toggleLight(0b01100100);
        _delay_ms(50);
        toggleLight(0b00100100);
    }

    PORTD = 0xff;
    _delay_ms(500);
    PORTD = 0x00;

}
void engineControl(void) //omtolkning av insignaler till motorsignaler
{

    if (recieve > 0 && recieve < 16) {
        switch (recieve) {

            case 1:
                Drive(0x1E);          //kör framåt
                break;

            case 2:
                Drive(0x16);          //kör vänster
                break;

            case 3:
                Drive(0x33);          //kör bakåt
                break;

            case 4:
                Drive(0x1A);          //kör vänster
                break;

            case 5:
                PORTD = 0x40;          //tänd bakljus
                Drive(0x12);          //stanna
                break;

            case 6:
                Drive(0x1B);          //snurra vänster
                break;

            case 7:
                Drive(0x36);          //snurra höger
                break;

```

```

        case 8:
            if(speed < 4)                //kolla så att hastigheten inte
understiger 1
            {
                break;
            }
            speed = speed - 3;           //minska hastigheten
            break;

        case 9:
            speed = speed + 3;           //öka hastigheten
            break;

        case 10:
            speed = 24;                  //sätt orginalhastighet
            break;

        case 11:
            blinkingLights(0b00001000); //blinkers höger
            break;

        case 12:
            blinkingLights(0b00010000); //blinkers vänster
            break;

        case 13:
            toggleLight(0b00100100);    //slå på strålkastare
            break;

        case 14:
            toggleLight(0b11111111);    //toggla alla lampor
            break;

        case 15:
            lightShow();                 //ljusshow!
            break;
    }
}

unsigned short int SPI_recieve(void) { //ta emot signaler från master

    while (!(SPSR & (1 << SPIF)));
    return SPDR;

}

```

```

void main() {

    SPI_slaveInit();                                //initiera SPI
    DDRA |= (1 << PA0) | (1 << PA2) | (1 << PA3) | (1 << PA5);
    PORTA |= (1 << PA1) | (1 << PA4);                //enable motors
    DDRD |= (1 << PD0) | (1 << PD2) | (1 << PD3) | (1 << PD4) | (1 << PD5) | (1
<< PD6);      //initiera lamporna
    direction = 0;      //sätter riktningen till 0
    speed = 24;          //anger starthastighet (delay)

    while (1)
    {
        recieve = SPI_recieve();
        engineControl();                                //skickar in mottagen
signal i enginecontrol

    }
}

```