## C-kod, kontrollen

```
#include <avr/io.h>
#include <inttypes.h>

#define CSN_LOW() PORTB &= ~ (1 << PB4); //slave select låg
#define CSN_HIGH() PORTB |= (1 << PB4); //slave select hög

unsigned short int transmit;

void USARTInit(uint16_t ubrr_value)                    //initiering av USART
{

  UBRRL = ubrr_value;
  UBRRH = (ubrr_value>>8);
  UCSRC=(1<<URSEL)|(3<<UCSZ0);
  UCSRB=(1<<RXEN)|(1<<TXEN);


}

char USARTReadChar()                                          //hämta information
{

  while(!(UCSRA & (1<<RXC)))
  {
                  //do nothing
  }
  return UDR;
}


unsigned short int USARTWriteChar(char data)    //skriv data till USART
{


  while(!(UCSRA & (1<<UDRE)))              //inmade bokstäver speglas tillbaka
via com-porten. En kommandosiffra skickas till bilen via SPI


  {
                  //gör ingenting
  }

  if(data == 'W')
        {
        UDR = 'W';
         return 1;
```

```
}
else if(data == 'A')
{
UDR = 'A';
return 2;
}
else if(data == 'S')
{
UDR = 'S';
return 3;
}
else if(data == 'D')
{
UDR = 'D';
 return 4;
}
else if(data == 0x20)
{
UDR = 'G';                          //mellanslagstangenten returnerar G
 return 5;
}
else if(data == 'E')
{
UDR = 'E';
 return 6;
}
else if(data == 'Q')
{
UDR = 'Q';
 return 7;
}
else if(data == '1')
{
UDR = '1';
 return 8;
}
else if(data == '2')
{
UDR = '2';
 return 9;
}
else if(data == '3')
{
UDR = '3';
 return 10;
}
else if(data == '4')
{
UDR = '4';
```

```
            return 11;
            }
            else if(data == '6')
            {
            UDR = '6';
            return 12;
            }
            else if(data == 'F')
            {
            UDR = 'F';
            return 13;
            }
            else if(data == 'P')
            {
            UDR = 'P';
            return 14;
            }
            else if(data == 'Z')
            {
            UDR = 'Z';
            return 15;
            }
            else
            {
            UDR = data;
            return data;
            }
}

void SPI_MasterInit(void) //initiering av SPI för master
{
            DDRB = (1 << PB4) | (1 << PB5) | (1 << PB7);
            DDRB &= ~ (1 << PB6);
            SPCR = (1 << SPE) | (1 << MSTR) | (1 << SPR0) | (1 << SPR1);
}


void SPI_transmit(unsigned short int transmit)
{

            CSN_LOW();                        //slave sätts till låg
            SPDR = transmit;        //sparar transmit på SPDR och startar överföringen
            while (!(SPSR & (1 << SPIF))); //väntar till lyckad sändning
            CSN_HIGH();               //slave sätts till hög
}


void main() {
```

```
	SPI_MasterInit();			//initiera master
	CSN_HIGH();
	char data;
	USARTInit(51);				//räknas ut UBRR-värdet mha
boadrate, bitstorlek och processorfrekvens
	while (1)
	 {
		data = USARTReadChar();
		transmit =  USARTWriteChar(data);	//läs av från USART,
skriv sedan på transmit
		if(transmit < 16 && transmit > 0)  //skicka endast vidare
styrkommandon via SPI
		{
		SPI_transmit(transmit);
		}
		else
		{
			//gör ingenting
		}
  }
}
```