Opsætning af mikrocontrolleren

Stack:

Stakken opsættes i starten af programmet til at stack pointeren peger på den højeste adresse i data Space. Dette sikres med brug af adressen med navnet RAMEND. Hvor den høje Byte bliver lagt i SPH og den lave i SPL.

Stacken bruges når fx en subrutine kaldes, så gemmes returnerings adressen i stacken. Stacken bruges også til midlertidig at gemme en værdi ved af flytte værdien til stakken og tilbage efter behov.

Hardware interrupt:

Der er blevet benyttet hardware interrupt til afstands måling og målstreg detektering. Til dette bruges INT0 og INT1. Da både afstands måling og målstregs detektering giver et faldende signal ved detektering sættes interrupt for dem begge med aktivering på faldende signaler. Derudover aktiveres pull-up for begge indgange

Opsætning af kommunikation:

Den serielle kommunikation opsættes til at have asynkron clock, ingen paritet, et stop bit og 8 bit besked størrelse. Derudover aktiveres modtagelse og afsending.

Der skal benyttes en baud rate på 9600 Bd. For opsætning af dette skal registret UBRR gemmes med en værdi udregnet ved hjælp af ligning \ref{eq:Kom1} givet i bogen ref. nummer ”The AVR Microcontroller and Embedded Systems: Usnig Assenmbly and C”

\begin{equation}

UBRR=\frac{Fosc}{16\cdot(Desired\:Baud\:Rate)}-1

\label{eq:Kom1}

\end{equation}

Fosc står for den clock-frekvens miccontroleren bruger ” Desired Baud Rate” er den ønskede baud rate.

Mikrocontrolleren bruger en clock-frekvens på 16 MHz og den ønskede baud rate er 9600 Bd.

\begin{equation}

UBRR=\frac{16\cdot10^{6}\:Hz}{16\cdot(9600\:Bd)}-1

\label{eq:Kom2}

\end{equation}

\begin{equation}

UBRR=103.167\approx103

\label{eq:Kom3}

\end{equation}

Dette giver at UBBR skal gemmes med værdien 103 \ref{eq:Kom3}for der bliver brugt en baud rate på 9600 Bd.

Da der bliver afrundet værdien til UBRR vil der forekomme en relativ fejl. Denne relative fejl kan udregnes med denne ligning \ref{eq:Kom4} fra samme bog på side 416.

\begin{equation}

Error=\frac{Calculated\:value\:for\:the\:UBRR\:-\:Integer\:part}{Integer\:part}

\label{eq:Kom4}

\end{equation}

Til udregning af den relative fejl \ref{eq:Kom5} benyttes resutater fra ligning \ref{eq:Kom3}

\begin{equation}

Error=\frac{103.167\:-103}{103}

\label{eq:Kom5}

\end{equation}

\begin{equation}

Error\approx 0.0016\approx 0.16 \%

\label{eq:Kom6}

\end{equation}

Så den relative fejl vil ligge på ca 0.16\% hvis dobbelt baud rate funktion benyttes ligger den relative fejl stadig ligger på ca 0.16\%.

Opsætning af PWM:

Timer 2 benyttes til at udføre PWM på motoren til bilen. Timer 2 er til dette sat til at være i ikke inverteret fasekorrekt tilstand. Den er sat til ikke at have prescaler grundet målinger som angivet her Ref:?????(De der målinger lavet af Jakob). Ved opsætningen bliver OCR2 gemt med 0 hvilket angiver en PWM på 0%. Timer 2 PWM udgange port D pin 7 sættes til at være en udgang.

Opsætning af ADC:

ADC er blevet aktiveret og sat op til at have interrupt sat til så resultatet af konverteringen vil blive behandlet når den er klar. Prescaleren er blevet sat til 128 grundet følgende udregninger.

Da ADC clock-frekvens skal være mindre end 200 kHz for at give præcise resultater, skal prescaleren vælges efter så dette opfyldes. Se side 475 i \cite{AVR}. Minimum krævet prescaler kan udregnes med følgende ligning \ref{eq:ADC1}.

\begin{equation}

Prescaler>\frac{F\_{ \tiny OSC}}{200\: kHz}

\label{eq:ADC1}

\end{equation}

Da miccontroleren bruger en clock-frekvens på 16 MHz er den minimale krævede prescaler:

\begin{equation}

Prescaler>\frac{16 \:MHz}{200\: k Hz}

\label{eq:ADC2}

\end{equation}

\begin{equation}

Prescaler>80

\label{eq:ADC3}

\end{equation}

Fra ligning \ref{eq:ADC3} ses den minimale presclaer er 80. Da den nærmeste prescaler der er over 80 er 128 vælges denne prescaler til ADC konvertering. Se side 475 i \cite{AVR}.

Som reference spænding til ADC konverteren er AVCC indgange blevet valgt da denne har en spænding på 5 V (….Noget med acc maks giver ?? V og der med denne ref spænding….). Resultatet fra ADC konvertering er blevet sat op til at være venstre justeret. Da dette er den nemmeste måde senere at få et 8 bit resultat i stedet for 10 bit da de 8 højeste bit er samlet. ADC konverterens indgang er valgt til at være ADC0. På denne pin er sat til at være et input og accelerometret er påsat denne.

Ref:

AVR side 475

Opsætning af RGB LED:

RGB LED’eren er sat på port a indgang 6 til 1. Alle indgange er blevet sat til output.

Opsætning af timer 1

Timer 1 bruges til at holde en bestemt LED værdi i et sekund. Til dette er timeren sat op til at være i CTC mode med en prescaler på 1024 og med interrupt ved overensstemmelse med OCR1A. Samt OCR1A gemt med værdien 15624. Denne værdi er fundet ud fra denne ligning \ref{eq:Time11}.

\begin{equation}

OCR1A =\frac{Tid\:[sek]}{\frac{Prescaler}{F\_{\tiny OSC}}}-1

\label{eq:Time11}

\end{equation}

Da miccontroleren bruger en clock-frekvens på 16 MHz og for OCR1A er under 0xFFFF hvilket er grænsen bruges prescaler på 1024.

\begin{equation}

OCR1A =\frac{1\:sek}{\frac{1024}{16\:MHz}}-1

\label{eq:Time12}

\end{equation}

\begin{equation}

OCR1A =15624

\label{eq:Time13}

\end{equation}

I ligning \ref{eq:Time13} ses at OCR1A skal gemmes med 15624 for timeren vil lave et interrupt efter 1 sekund fra den er startede.

Fuse bit

For at EEPROM hukommelsen ikke bliver slettet ved programmering af miccontroleren er EESAVE bitet i Fuse Byte High blevet ændret fra 1 til 0. Se side 294 i \cite{AVR}. Udover dette benyttes et fuse givet af lektor Karsten Holm Andersen Mærsk Mc-Kinney Møller Instituttet SDU. Denne fuse gør blandt andet at miccontroleren benytter det externe 16 MHz krystal.

Ref:

AVR side 294