Aarhus universitet

Anvendte microcontroller systemer

6. Semester

Color Sorting System

Gruppemedlemmer:	*AUID*
Daniel Tøttrup	au544366
Stinus Lykke Skovgaard	au520659

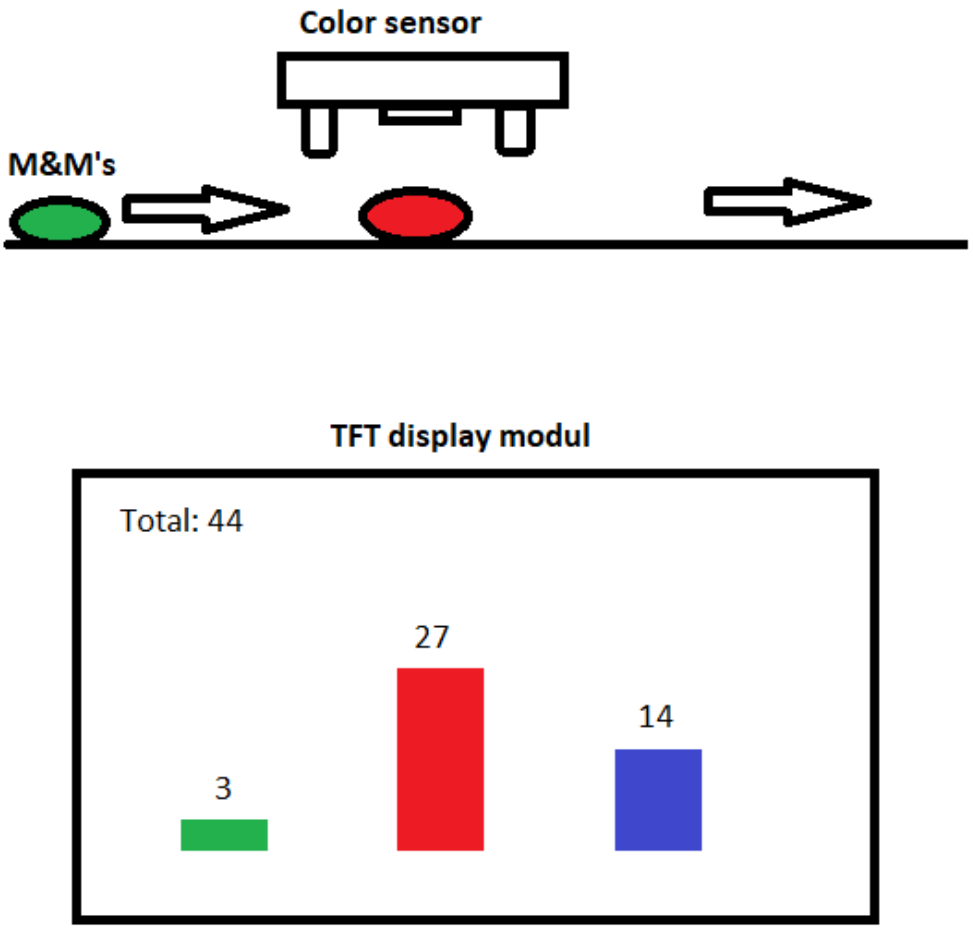
AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

26. maj 2018

Indhold

Figurer

1 Indledning



Figur 1: Konceptbillede for CSS

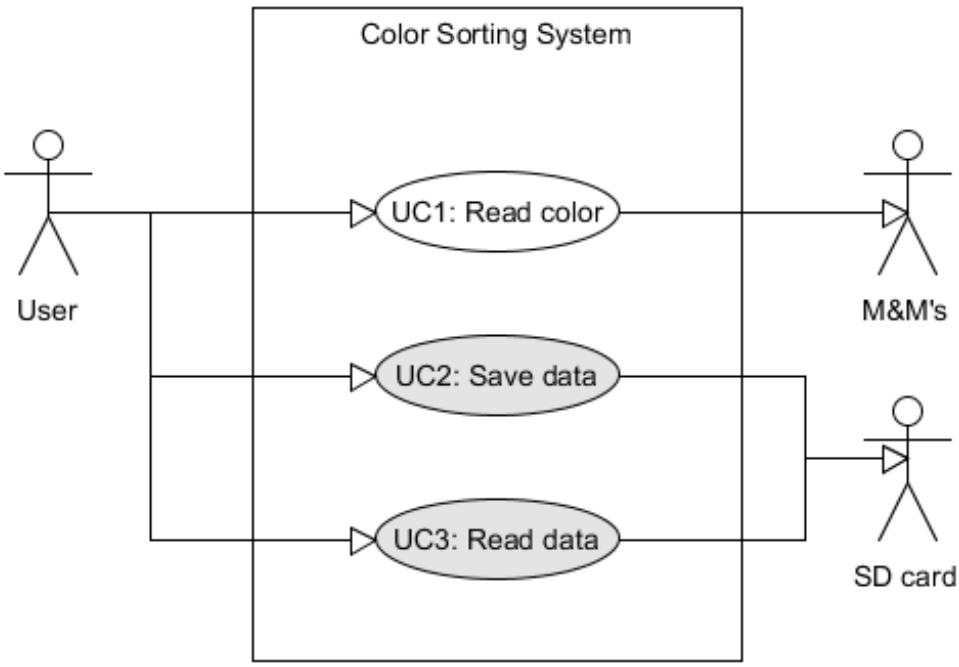
Color Sorting System(CSS) gør det nemt at sortere diverse emner baseret på deres farve. Disse emner kan være alt fra fødevare til maskin-komponenter, så længe de kan identificeres på farve. CSS fungerer ved at lade emnet passere under en farve sensor, som kan identificere farven på emnet og videresende hvilken farve emnet har til et TFT displaymodul. TFT modulet kan så ved hjælp af søjle diagrammer, fortælle hvor mange forskellige farvet emner der har passeret sensoreren. Resultaterne kan gemmes på et SD kort, hvis man senere skulle bruge dataen. Dog er implementeringen af SD kortet ikke fuldendt i denne prototype, og vil i en viderudvikling af projektet have stor prioritet. Desuden er der kun lagt vægt på color sensor og TFT display modulerne, hvilket vil sige at selve sorteringsmekanikken ikke er implementeret i den nuværende prototype.

2 Krav

I dette afsnit beskrives kravene til CSS og hvilken funktionalitet systemet skal have. Der er stillet nogle enkelte krav fra undervisers side, hvor nogle af disse krav skal indgå i projektet:

- Use In Circuit debug tools
- Implement Drivers, dealing with time critical parameters.
- Implement Boot Loaders for updating microcontroller firmware
- Use USB to interface a microcontroller
- Use operating systems for microcontrollers
- Use microcontroller knowledge in a final mini project

Flere af disse krav er implementeret i CSS. Der er brugt tidskritiske drivers til color sensoren og til implementeringen af I2C mellem de to microcontrollers.



Figur 2: Usecase diagram for CSS

På Figure 2 ses usecase diagrammet som beskriver sammenhængen mellem aktørerne og de forskellige funktionaliteter der findes for systemet. Uscase 2 og 3 er grå, hvilket betyder at de ikke er implementeret i prototypen.

2.1 UC1 - Read Color

Denne usecase danner ramme for hvordan CSS måler en farve. Brugeren placerer emne der ønsket aflæst. Brugeren trykker på aflæs-knappen og den aflæste farve kan nu ses talt op på TFT display modulet.

2.2 UC2 - Save Data

Denne usecase danner ramme for, at gemme data på SD kort. Brugeren trykker på "Save Data"knappen på skærmen. Data bliver derefter gemt på SD kort.

2.3 UC3 - Read Data

Denne usecase danner ramme for, at hente data fra SD kortet. Brugeren trykker på "Load Data"knappen på skærmen. Gemt data bliver hentet fra SD kortet og vist på TFT skærmen.

2.4 Afgrænsning

Det skal sige at usecase 1 og 2 ikke er implementeret i denne protorype. Der er gjort forsøg på at få dem implementeret, men grundet tidspres fik de Derudover var det også tænkt fra start, at der ikke skulle være behov for brugerinput for at kunne aflæse farve, men på grund af tidsmangel fik gruppen ikke implementeret et system der kunne fodre CSS med emner. Dette gøres istedet for manuelt og vil i fremtiden skulle noget automatisering af CSS implementeres.

3 Color Sensor Module

Color sensor modulet indeholder både en microcontroller og en color sensor. Til dette projekt har gruppen valgt at bruge en LC Technology TCS3200. Sensoren blev valgt fordi den var på lager i Embedded Stock, og at den ville passe godt til dette projekt. Til at styre denne sensor bruges en Arduino mega 2560.

3.1 LC Technology TCS3200

LC Technology TCS3200, virker ved at have 8x8 array af fotodioder. 16 fotodioder med et grønt filter, 16 fotodioder med et blåt filter, 16 fotodioder med et rødt filter og 16 fotodioder uden noget filter. De kan alle styres ved at sætte to pins(S2 og S3) høj eller lav. Se Figure 3

S2	S3	PHOTODIODE TYPE
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Figur 3: Styring af photodioder

For at aflæse farveintensiteten, bliver man nødt til at måle på sensorens output pin. Signalet der kommer ud er et firkantssignal og farveintensiteten er bestemt alt efter hvor høj frekvensen er. For at se hvilken farve emnet har, bliver man nødt til at aktivere de forskellige fotodioder hver for sig, og tage en måling på output pin'en hver gang man har skiftet fotodiode. Derefter kan man så sammenligne de tre målinger (Clear bliver ikke målt) og se hvilket er størst. Hvis der skal måles andre farver end rød, grøn og blå, som fx gul, bliver man nødt til at kigge både på grøn og på rød. Hvis begge er lige høje må det være gul. Dog reagerer de forskellige fotodioder forskelligt på hvilken farve man ser på, så der skal der tages højde for. På Figure 4 kan man se hvordan de forskellige fotodioder reagerer ved forskellige bølgelængder(farver).

PARAMETER	TEST CONDITIONS	CLEAR PHOTODIODE S2 = H, S3 = L			BLUE PHOTODIODE S2 = L, S3 = H			GREEN PHOTODIODE S2 = H, S3 = H			RED PHOTODIODE S2 = L, S3 = L			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_O Output frequency (Note 9)	E_e = 47.2 µW/cm², λ_p = 470 nm	12.5	15.6	18.7	61%		84%	22%		43%	0%		6%	kHz
		(4.7)	(5.85)	(7)										
	E_e = 40.4 µW/cm², λ_p = 524 nm	12.5	15.6	18.7	8%		28%	57%		80%	9%		27%	
		(4.7)	(5.85)	(7)										
	E_e = 34.6 µW/cm², λ_p = 640 nm	13.1	16.4	19.7	5%		21%	0%		12%	84%		105%	
		(4.9)	(6.15)	(7.4)										

Figur 4: Output frekvens ved forskellige farver

En sidste ting der er værd er at vide om TCS3200, er dens indbygget frequency scaler. Den kan bruges til at styre skaleringen af output signalets frekvens. Skaleringen kan styres ved

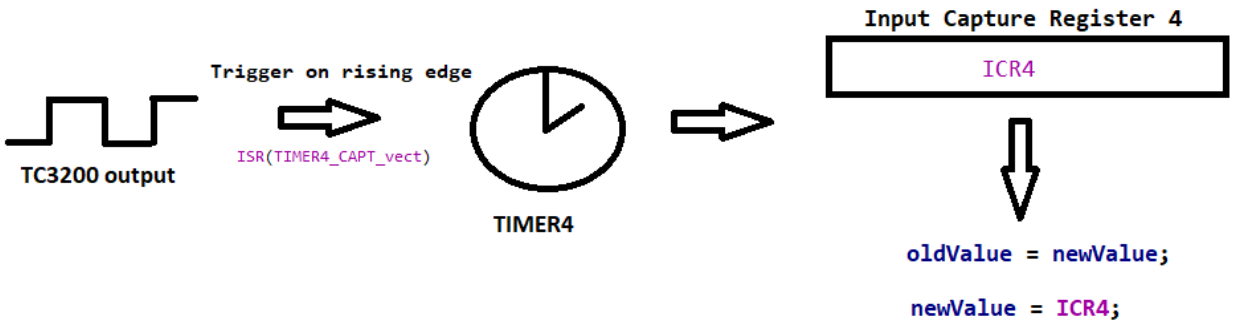
at sætte S1 og S2 høj eller lav. Til CSS bruges 2% skalering, da 100% skalering, ville betyde at output frekvensen kunne komme op over 50kHz. En lavere frekvens vil være fordelagtig i forhold til at få en præcis måling. Se Figure 5

S0	S1	OUTPUT FREQUENCY SCALING (f_o)
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

Figur 5: Output frekvens skalering

3.2 Input Capture

Input Capture er en metode der ofte bruges i embedded systemer, til at måle på diverse signaler. Til at måle outputsignalet fra TC3200 color sensor bruges denne metode.



Figur 6: Input Capture illustration

Input Capture fungerer ved at have et interrupt der trigger på rising edge. Når dette interrupt bliver kaldt, tages et øjebliksbillede af TIMER4s værdi, som gemmes i Input Capture Register 4. Denne værdi gemmes i en variabel, som bagefter bruges til at beregne en frekvens.

Derudover skal der også tages højde for overflow, ellers kan man risikerer at en måling ikke vil give en korrekt frekvens. For at udbedre dette problem bruges en anden interrupt, ISR(TIMER4_OVF_vect). Den trigger hver gang der kommer overflow, og 65535 lægges til newValue. Koden til hvordan frekvensen udregnes kan ses nedenunder.

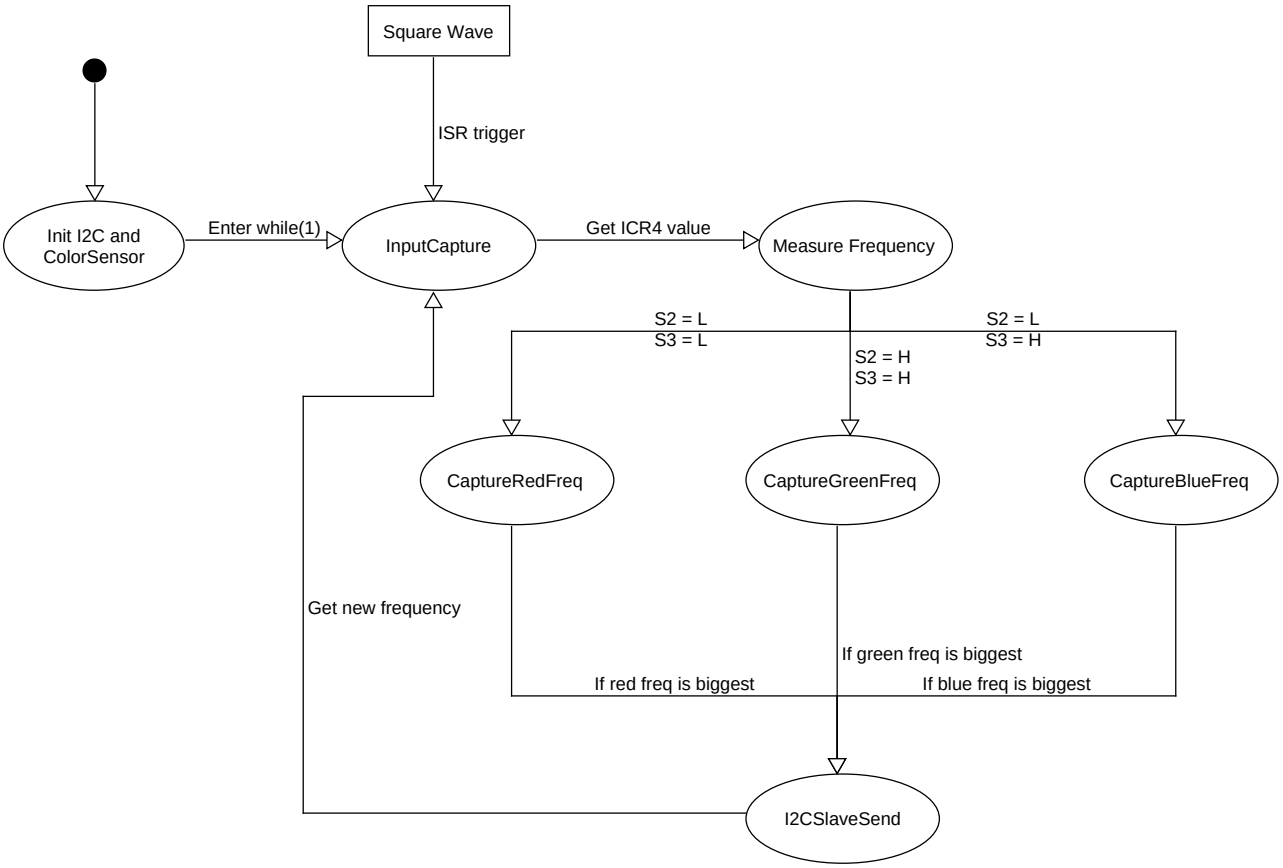
```
ISR(TIMER4_CAPT_vect)
{
    oldValue = newValue;
    newValue = ICR4;

    if(newValue < oldValue)
    {
        period = oldValue-newValue;
    }
    else
    {
        newValue + overflow;
        period = oldValue - newValue;
    }
    freq = F_CPU/period;
    FREQFLAG = 1;
}
```

Det kan også ses i koden, hvordan overflow værdien lægges til newValue, hvis newValue er mindre end oldValue. FREQFLAG bruges så man altid er sikker på at freq har en ny værdi. FREQFLAG skal selvfølgelig sættes til 0, når man har brugt freq.

3.3 Color Sensor Module Software

For nemt at kunne forstå softwaren brugt til TC3200, er der blevet udarbejdet et data flow diagram, der giver et overblik over dataflowet i softwaren. Dette diagram kan ses på Figure 7. Som det ses i diagrammet, bliver der taget en frekvensmåling for hver farve. Dette gøres, som beskrevet tidligere, ved at sætte S2 og S3 enten høj eller lav. Når en måling er taget, sammenlignes de tre målinger for at se hvilken farve er mest repræsenteret. Derefter sendes en char som enten er 'R', 'G' eller 'B'. Kommunikationen sker via I2C. Derefter starter koden forfra igen, og tager en ny frekvensmåling.



Figur 7: Data flow diagram

4 Struktur

5 Test

6 Diskussion

7 Konklussion

Litteratur

Litteratur