Opgave 3

Formålert:

1. At genere en Pulse Width Moduleret (PWM) signal med Timer A1 på P2.0 med frekvens 1 kHz og duty-cycle bestemt med 8 bit dip-switch, – højeste decimal værdi er 255 eller i hex 0xFF. Hvis der er behov kan switch værdierne efter de er læst ind ganges med 2, 4, 8.

2. At måler på PWM signalet før filter (PWM_out og efter filter DC_out vha. et oscilloskop).

3. At prøve med forskellig filter afskæringsfrekvens for at minimere ripplen på det filtrerede output Vælg et RC led og vælg modstanden til 10 kohm eller større, afskæringsfrekvens 100 Hz, dvs. beregn så C. Eksperimenterer lidt med C værdien

4. At få opsat en digital analog konverter ADC12 i msp430F55229, med en passende samlerate på ca. 1000 gange i sekundets evt. styret vha. en timerA0.

5. At få interrupt på ADC12 complete sat op, så der er interrupt, når sample er klar i hukommelsen.

6. At benytte den digitale repræsentation fra ADC til at styre pulsebredden vha. den indbyggede analog til digital converter ADC12, hvor ADC12 styres vha. af en variabel spænding sat til ADC12 ’s indgang.

7. Efter filteret (DC_out) at prøve at belastee PWM signalet og kompensere fladet i DC værdi ved at justere indgangsspændingen på indgangen af ADC-converteren.

8. At udføre en P regulering dvs. ved at forbinde indgange af analog digital converten til det filtrede pwm signal. Og at belaste og måler spændingen over filteret.

43	**24-10**	**6**	**Clock kilder i MSP TimerA som pulse width modulator (PWM) opsætning af timer1A’s PWM signal på P2.0 og filtreret –måling af pulsevide og ripple**	**s.163-167 s. 330- 352 (øverst)**	**TimerA_user_guide_slau400f.pdf: s 2-22**	**Opgave 3 1-3**
44	**31-10**	**7**	**Analog digital konvertering – sample rate opsætning af ADC converter i c – adc styring af pwm**	**s. 393 – 403, s 432-438**	**msp430f5529: s. 31 ADC12_A_user_guide.pdf: s 2-7, s.9, s10, s.12, s. 14, s.21-31**	**Opgave 3 4-5-6**
45	**7-11**	**8**	**DC spænding styring: PWM filtreret output med belastning – proportional kontrol af pwm out – med filtreret pwm som feedback**	**s. 424 -427**	**TimerA_user_guide_slau400f.pdf: s 2-7-øv, s. 17-22**	**Opgave 3 6-7**

Dokumentation

Der skal udarbejdes en journal med forklaringer til de forskellige opsætninger i microcontrolleren for PWM, ADC og timeren der styrer sampleraten. Interrupt diagram hvor man kan se de forskellige interrupt service rutiner, relevante flow charts og test resultater målt vha. at oscilloskop målinger og billeder af disse. Dokumentation for målt pulsebredde (på PWM_out) passer til beregnet pulsbredde (duty Cycle). Skriv timers TA1R værdi ud på OLED display, TA1CCR1-værdi ud på display- og ADC12 resultat ud på display og se den varierer når spændingen på indgangen af ADC12 varieres – tag billeder af det. Pkt 7 og 8 dokumenteres med billeder af målinger vha. oscilloskopet og notation af spændinger i bilaget til opgaven. Åben et fælles dokument op på google docs eller i teams – noter og dokumenter med billeder og tekst

Hint’s vil være vha. bogen, slides til forberedelse og i lektionen og dette dokument

Tidsplan:

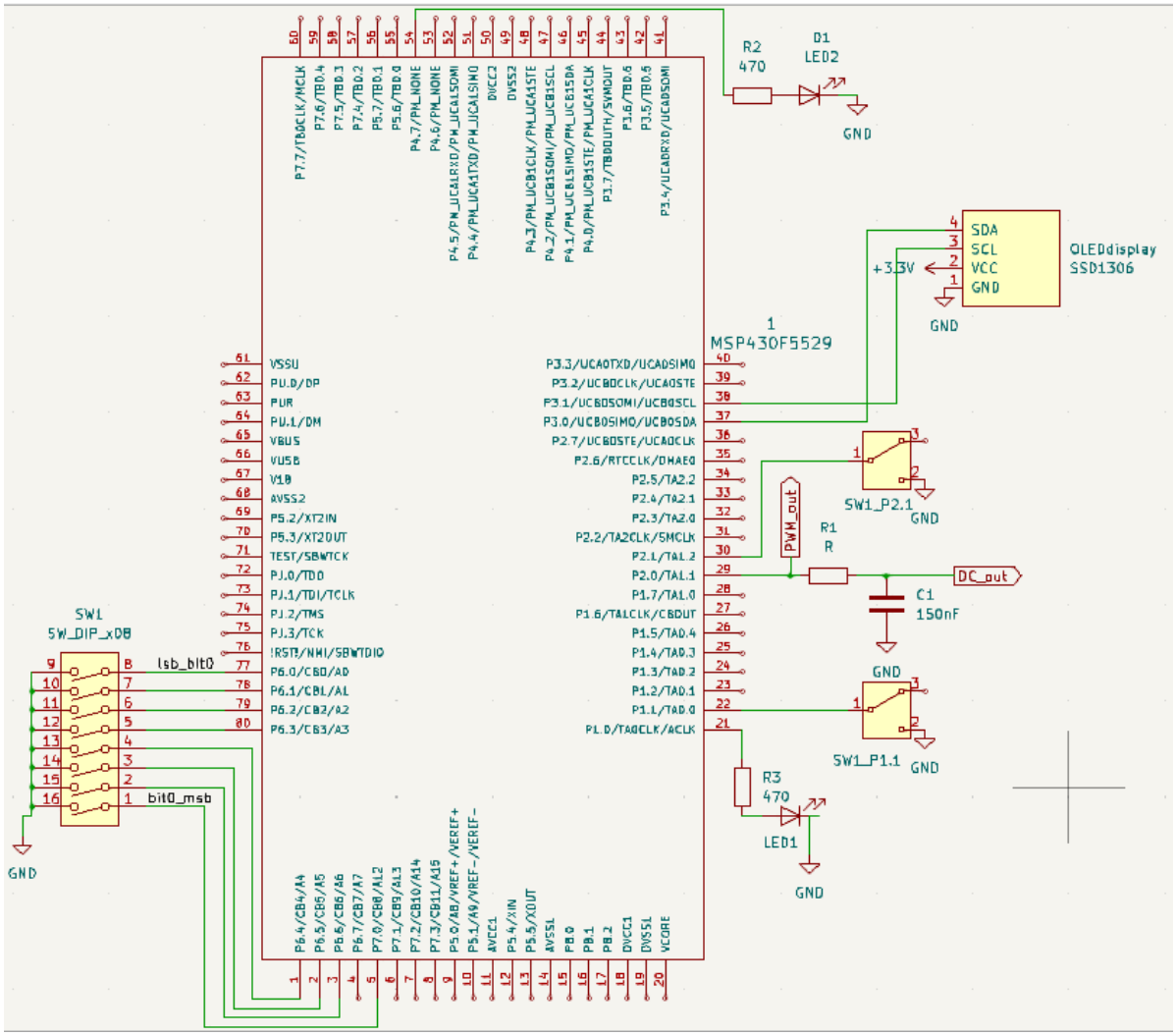
Parallelt med opgaven gives undervisning der matcher punkterne typisk 2 til 3 punkter per uge. – jfr. kursusplanen.

DVS. I skal for hver uge have udført arbejdet - der vedrører lektionens indhold inkl. Forberedelse.

På learn under opgaven er udsnit af dokumentationen for Timere timerA1 og ADC12 er vedlagt.

Aflevering 12. nov. kl. 23.59 af zippet main.c og filerne adc.h og adc.c og et pdf dokument med dokumentation per gruppe.

Diagram for opstillingen til at svare på 1, 2, 3 vises her. Det såkaldte RC filter ses tilsluttet P2.0 hvor PWM_out kommer fra timer A1.1 – Display og dipswitch skal forsat være forbundet!



HINTS

1. DEL

Ad. 1

Eksempel kode fra lektion 6 kan benyttes med lidt ændringer. Ad 1-2. Timer A1 sættes op til at genere centeret PWM – dvs. timer skal sættes til, tælle up mode for at få edge alligned PWM generet. Initialiser TimerA1 til at genere en edge aligned PWM med 4 MHz SMCLK clock frekvens (f_{source}) ind i timerA1 (ingen neddeling af klok-

frekvens). Og maks. værdi/ top værdi for tæller sættes med compare register TA0CCR0 før den nulstilles og værdien skal være 4095. Udregn PWM frekvensen vha. formlen i regnearket eller i slide-serie for lektion 6.

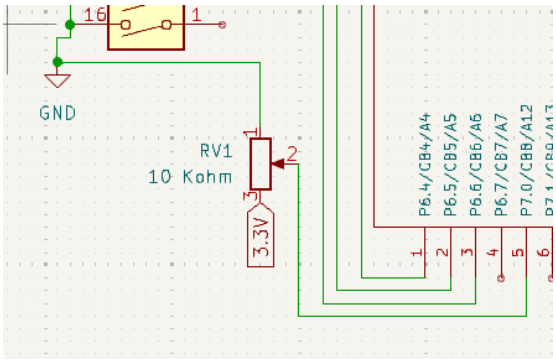
Som udgangspunkt ønskes en duty cycle på 50%. TA1CCR1 sammenlignings-register bestemmer sammen med top-værdien duty cyclen så beregn en værdi for TA1CCR1, så der fås 50% duty cycle vha dette udtryk: Hvilken værdi skal TA1CCR1 have i forhold til TA1CCR0? Værdien i TA1CCR1 skal sættes vha. dipswitch – men den indlæste værdien skal ganges med 16 før den skrives ind i TA1CCR1 (da 4095 er en 12 bit maksværdi- så 4 bit mere dvs. 2^4 gange større end de 8 bit der er i dip-switch). Den skal læses ind når **TA1CTL&TAIFG** er sand, dvs. når tælleregister er ved topværdi, så sættes et Timer A1 interrupt flag TAIFG i register TA1CTL. Verificere at duty-cycle er 50% med måling på Oscilloskop –mål på P2.0 (pwm_out). Video om måling med oscilloskop er bilag til opgaven

Skriv TA1CCR1's værdi ud på display og se at det passer med at tallet er 16 gange større end det der er sat på dip-switch. Man kan vha. funktionen `ssd1306_printUI32( uint8_t x, uint8_t y, uint32_t val, uint8_t Hcenter ) skrive heltal ud af typen uint32_t` uden at skulle benytte konvertering til ascii chars med sprintf. For udskrivning blot benyt selve registernavnet (ex. `ssd1306_printUI32(`0,1, TA1CCR1,2); Så I programmet i main udregnes dutycycle og skriv den ud på display. Hvor godt stemmer oled-display duty cycle med den målte? Prøv at regne en ny duty cycle ud f.eks. 75%, læs den værdi ind fra dip-switch gange med 16 der vil give 75% duty cycle. Mål med oscilloskop. Hvor godt stemmer det?

Ad. 2 og 3. Vha. slide serie lektion 6 kan der udregnes afskæringsfrekvens $f = \frac{1}{2pi*R*C}$ for at minimere/midle PWM pulserne, så man tilnærmelsesvis får en dc spænding (efter filter- DC_out), hvis amplitude er styret af pulsbredden. Afskæringsfrekvensen skal sættes mellem 50 til 100 gange lavere end PWM-frekvensen. Prøv at ændre pulsbredden ved at indlæse værdien fra dip switch ind i en variabel – gang/skaler denne variable med 16 (hvorfor det?) og sæt denne skalerede værdi ind i sammenligning register TA1CCR1 – hvor i skifter fra 10% til 90% duty cycle. Læs duty-cyclen ud på OLED display'et. Tag nogle billeder af oscilloskopets skærm med forskellige filter frekvenser og duty cycles PWM_out målt på ch 1 og efter filter DC_out på ch2.

Del 2:

Ad. 4. Få initialiseret ADC12 så den kan sample analoge spændinger på A12 (P7.0) - Derfor fjern dip-switch benets tilslutning til P7.0 Hent et potentiometer på 10 kohm eller en trimme-modstand og forbind den til 3.3 v og gnd og tag midter-benet og forbind til P7.0 som vist her:



Opret et nyt projekt i visual studio insiders og Initialiseret ADC12. Benyt ADC12CLK og configurer ADC-mode til single-channel mode – se adc initialiserings funktion i slideserien for lektion7-E24 og ret eksemplet så kanal A12 benyttes (dvs. at vi kan få det analoge signal ind på P7.0) – check med dokumentation for ADC12 (bilag til opgaven), at de øvrige bit er sat rigtigt.

I slide serien er en polling funktion som kan benyttes til at læse data fra adc – men **før** get_sample funktion kaldes, så skal adc startes med `ADC12CTL0 |= ADC12ENC | ADC12SC;`

I kan benytte oled display til at læse den digitale repræsentation N_{ADC} for spændingen ud. Og med et voltmeter mål hvad spænding er på P7.0 og benytte denne formel til udregning af N_{ADC}

$$N_{ADC} = 2^N * V_{in}/V_{FS}$$

Til at udregne N_{ADC} hvor V_{FS} sættes til 3.3V og N= 12 bit. Det digital repræsentation af dc spændingen er N_{ADC} og værdien er i register ADC12MEM0. Værdien kan udskrives ved at kalde denne funktion OLEDssd1306 ssd1306_printUI32(uint8_t x, uint8_t y, uint32_t val, uint8_t Hcenter) (ex. ssd1306_printUI32(0,0,ADC12MEM0,0); Sammenlign den udregnede værdi for N_{ADC} med den udlæste ADC12MEM0 værdi på OLED-display.

Ad. 5 Når det virker så enable interrupt på adc12 og service rutinen sættes ind – Erklær to globale variable et adc_flag og en til at holde den konverterede adc-værdi f.eks. kald den data. Også dette fås fra slide serien lektion 7 -I service rutinen sættes en variable adc_flag=1 til 1 og læs ADC12MEM0 over i data variabel. Og i main testes på dette adc_flag med et if – hvis adc_flag lig 1 skriv data variablen ud med ssd1306_printUI32 funktionen, sæt adc_flag=0; og efter et delay på __delay_cycles(40) – det giver et delay på 1 ms, når SMCLK er 4MHz og det betyder der samples 1000 gange i sekundet, derefter skrives ADC12CTL0 |= ADC12ENC | ADC12SC; for starte ny sample. Test at det virker igen med voltmeter og udregning af den digital værdi og sammenlign med det der står på OLED display.

Option er at styre sample raten i vha. Timer TA0 comparematch interrupt som I arbejdede med i opgave 2 – timeren skal sættes op til at interrupte 10000 gange i sekundet og i service rutinen så sættes ADC i gang med at sample med dette: ADC12CTL0 |= ADC12ENC | ADC12SC; se eksempel i slide lektion 7/8

Ad. 6 Opret et nyt projekt i visual studio insider læg koden fra lektion 6 og 7 ind. Benyt passende funktioner til at omkranse de enkelte enheders initialisering, eks. void adc_init(), void display_init(), void pwm_init() og/eller lav moduler .c og .h moduler for adc converter koden og pwm koden -så kan de let genbruges i opgave 4!

Nu skal adc værdien fra ADC12MEM0 register skrives ind i PWM timeren sammenligningsregister TA1CCR1 register i main – Det sker blot ved at i main når adc_flag=1 og **TA1CCR1=ADC12MEM0**. Compiler og test at det virker ved at måler på P2.0 udgangen med oscilloskopet før filter og skrue på potentiometeret mens – så skal i kunne se pwm duty cyclen ændrer sig på oscilloskopet med potentiometers position. Svar på hvorfor et det praktisk at topværdien bestemt med TA1CCR0 er sat til 4095 i forhold til muligheden for at få 100% duty cycle når spænding ind på adc er 3.3 V?

Dokumenter med passende oscilloskop billeder – voltmeter visning og forklar med tekst hvad vi ser.

Del 3

Her er målet at genere en konstant dc-spænding der kan modstå belastning ved en lukket sløjfe regulering så man med ADC konverter måler fejlen på udgangen DC_out i forhold til en ønsket værdi sat i programmet.

AD 7. Slides for lektion 8 er relevante. Der oprettes et nyt projekt og koderne fra lektion 7 lægges ind.

Potentiometeret midter pind er forbundet til P7.0 og yder ben er henholdsvis forbundet til +3.3 og gnd. Sæt spændingen ud af potentiometer midter pind **til 1 V** og mål det med oscilloskop - noter ned og i koden sæt adc12 resultat-værdien fra ADC12MEM0 ind i timerA1 sammenligningsregister TA1CCR1. Læsning af ADC

resultatet foregår som under ad4 eller ad5 ovenfor. Mål duty-cycle med oscilloskopet og pwm frekvensen samt dc-spændingen efter filter DC_out. Dokumenter det med billeder inkl. forklaringer

Belast nu med 100 kohm og mål igen. Nu forbindes en 100 kohm modstand parallelt over kondensatoren. Duty cycle, pwm frekvens og dc-spænding efter filter (DC_out) målers og noteres ned. Og prøv så at dreje på potentiometer så spænding bliver 1 v igen efter filter. Mål spændingen ind fra midter pinden af potentiometer. Hvor meget større er den? Med oscilloskopet mål duty cycle med oscilloskopet på P7.0 (PWM_out). Hvad er forholdet mellem den nye spænding ind og den tidligere spænding uden belastning? Hvad er forholdet mellem duty cycle før der belastes med 100kohm og efter der er kompenseret på indgangsspændingen så der stadig er 1 V over de 100kohm. Prøv at ændre kondensator og modstand så C=1.4uF og modstand er 10kohm og prøv at måle igen med 100 kohm belastning

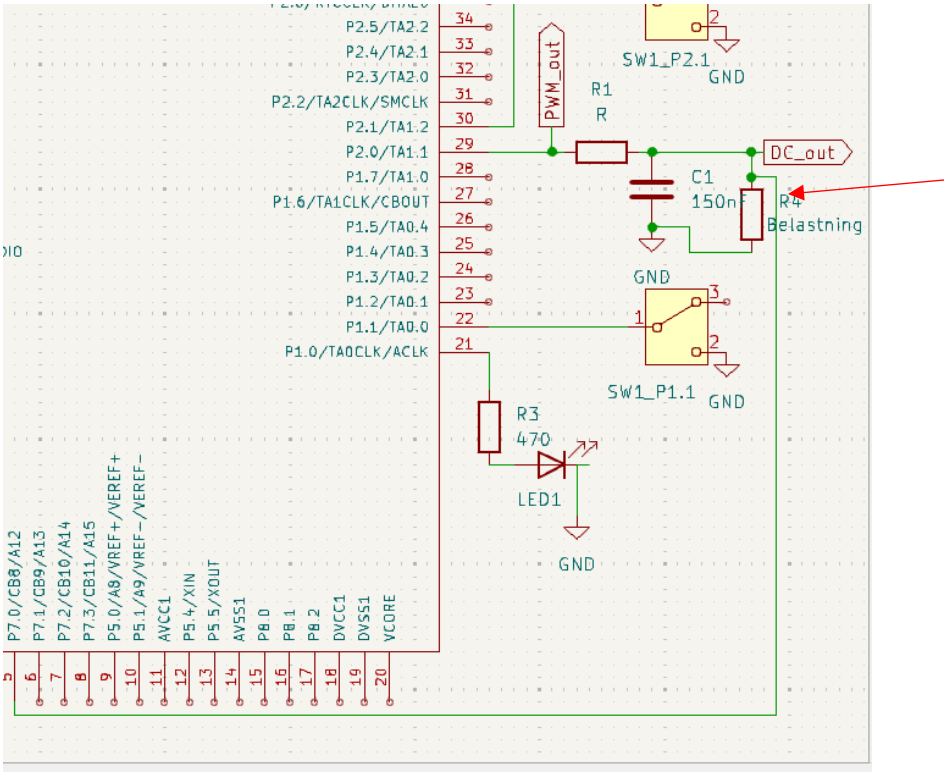
Skema til indsætning af værdier

Filter R	Filter C	belastning	Spænding ind	PWM dutycycle	PWM frekvens	Spændingen efter filter		
100k	150n	ingen						
100k	150n	100kohm						
			Kompenseret sp ind				Spænding ind komp/spænding ind	PWM duty med komp/pwm duty cycle
10k	1,5u	100 kohm	Kompenseret sp ind				Spænding ind komp/spænding ind	PWM duty med komp/pwm duty cycle

Dette er open loop kompensation manuelt med potentiometer.

ad 8 lukkes og adc12 måler på spændingen efter filter.

Ad. 8 Så potentiometer fjernes og der sættes en ledning fra DC_out til P7.0 her:



Dette fuldender feedback-sløjfen! Den ønskede dc-værdi efter filter vil nu blive defineret af en variabel Vd (V demand) i programmet. Med dipswitch sættes ønskes digitalværdi for spænding på 1 V, - udregn den digitale værdi er som ovenfor ad 5 og sæt dipswitch til den værdi, dvs. Vin er 1 V og Vref er 3.3 og n= 8*16 bit! Da dip switch kun har 8 bit (men det mest betydende bit skal benyttes til adc som ovenfor) derfor skaleres op med 4 bit

Mål så spænding over filter **kondensator uden belastning** af R4 og find G baseret på måling – (G i program sættes til 1)

I main definer også globale (float) variablerne G (loop_gain, Ve (digital_error_volts), Vd (digital_demand_volt) Vo(digital_open Circuit volt) og evt. D (duty_cycle i %).

Indsæt i main() under betingelsen når adc_flag=1 og **TA1CTL&TAIFG** Her udregnes så digital_error_volts. Den værdi skal testes for overflow dvs. hvis værdien er større end 4096, så skal den digital_error_volts sættes til digital_vd og TA1CCR1 = TA1CCR1+ digital_error_volts. Det betyder hvis afvigelsen er nenativ så træøkkes fejlen fra TA!CCR1 og det stemmer med at pulsebredden så skal sættes ned for at Vo svarer til Vd. jfr slide serie lektion 8 opdateret.

Vigtigt: Sørg for at forstå sammenhængen mellem disse variabler – se også slides.

Forbind oscilloskop-proberne til (a) P7.0 PWM_out filterindgangen ch1, (b) filterudgangen DC_out ch2. Kør koden og eksperimenter med forskellige Vd-indstillinger (spændings efterspørgsel) og forskellige værdier af G (løkkeforstærkning), og observer, hvordan udgangsspændingen ændres, når belastningsmodstanden tilsluttes.

Du bør se den store fordel ved at bruge feedbackkontrol!

Dokumentation:

- Vis C-koden og oscilloskopbillederne for hver af disse to situationer med forklaring (4 skud i alt):

(a) hvor efterspørgslen (Vd) på 1V gør det muligt for systemet automatisk at kompensere for virkningen af belastningen (2 skud, et med belastning afbrudt og et med belastning tilsluttet og kompenseret) og

(b) hvor kravet (Vd) gør det umuligt for systemet automatisk at kompensere for virkningen af belastningen (2 skud, et med belastning frakoblet og et med belastning tilsluttet og et mislykket forsøg på at kompensere).

Prøv at skifte R1 til 10 kohm og kondesatoren til 1.5uF og belast igen med 100 kohm. Kan regulatoren med passen G så opfylde at spændingen på Vo er det samme som uden belastning

Filter R	**Filter C**	**Chosen**	**Chosen**	**Chosen**	**Calculated**	**Calculated**	**measured**	**Mea osci**
100k	**150n**	**Demand Vd**	**Load (∞)**	**Loop gain G**	**G/(1+G)**	**$G_{cl}/(1+G_{cl})$**	**Duty cycle D**	**Out Vf**
100k	150n		uendelig					
			100kΩ)					
10k	1.5u							
			uendelig					
			100 k					

* Når belastningen er tilsluttet, skal sløjfeforstærkningen G korrigeres til Gcl, hvor Gcl = G*Gl

hvor Gl er forstærkningen af belastnings sektionen (her mindre end 1) på grund af spændingsdeleren bestående af belastnings modstanden og filtermodstanden.

Kommenter resultaterne – er de som forventet????