

MALMÖ HÖGSKOLA

Inbyggda system och signaler
Styrteknik

Inlämningsuppgift 1501b

Utlämning: 2015-11-9
Deadline: 2015-11-16

Namn: Leonard Holgersson

Namn: Hadi Deknache

Gion Koch Svedberg
november 2016

Sekvensstyrning med Matlab och Arduino Due

Syftet med denna inlämningsuppgift är att lära känna och använda utrustningen som används i kursen för styrningen och regleringen av olika processer. Första inlämningsuppgift handlar om Matlab, Arduino Due, Motor-shield och hur det används för att läsa in sensorvärden och styra motorer i olika tillämpningar.

Anvisningar

Skriv inte ut detta dokument utan ha det öppet på datorn och besvara frågorna direkt i dokumentet. Det är viktigt att ni följer denna anvisning noggrant i rätt ordning steg för steg, annars finns det risk att ni missar viktiga delar och behöver börja från början eller att utrustning blir förstörd!!

Inlämningen av detta fullständigt ifyllda dokument samt andra filer som ni ska generera för att dokumentera vissa delar av er lösning ska ske på its learning.

Uppgiften genomförs som vanligt i par dvs. ni jobbar två och två. Vid inlämningen på Its learning och på detta dokument titelsida anges vem som jobbat ihop. Forskningen visar att den mest effektiva inläringen sker när man förklarar något till någon annan! Tillämpa det gärna på varandra i gruppen och i hela klassen för att få hjälp i att förstå vad som ska göras och varför. Själva laborationen blir dock meningslös om ni fuskar och bara kopierar varandras resultat! Varje grupp ska ha egna svar, programkod och förklaringar!!

Ni behöver dessutom en **digitalkamera eller en mobiltelefon med kamera** för att kunna dokumentera era resultat.

Krav för godkänd

- Fullständigt ifyllt dokument med korrekta, egna svar och kod till alla frågor, uppladdad som **PDF-fil** på its learning
- Bildfil från uppgift 1.1.3, uppladdad till its learning
- Excelfil med resultat från uppgift 4.3, uppladdad till its learning
- Inspelad video av fungerande styrning av "slutuppgift": antingen som länk till ett eget youtube klipp eller som videofil (obs helst max 30 sec video), uppladdad till its learning

Inlämningsuppgifter som lämnas in innan deadline blir granskade av läraren som återkopplar med skriftliga kommentarer, beroende på rapportens utförande:

- Rapporten blir godkänd
- Komplettering krävs
- Rapporten blir underkänd

Efter eventuell komplettering som görs under kurstillfället:

- Rapporten blir godkänd
- Rapporten blir underkänd

Inlämningsuppgifter som lämnas in efter deadline men under kurstillfället blir granskade av läraren som återkopplar med skriftliga kommentarer, beroende på rapportens utförande:

- Rapporten blir godkänd
- Rapporten blir underkänd

I händelse av underkänd rapport eller om komplettering inte inkommer under kurstillfället ges nya inlämningstillfällen i anslutning till omtentamenstillfällena. Vid dessa ges dock inga kompletteringsmöjligheter, dvs rapporten blir godkänd eller underkänd.

Lägg märke till att laborationsutrustningen normalt endast finns tillgänglig under kurstillfället!

Lyckas man inte få godkänt för alla labbar under kursens gång, måste alla labbarna göras om nästa gång kursen ges igen! Anledningen är att vi inte kan tillhandahålla utrustningen utanför kursen samt att vi kontinuerligt anpassar och förändrar labbarna i våra kurser.

Tillgängliga referenser på its learning

- [1] Mathworks, Matlab primer, R2015b
- [2] Sune Bergelin, Örebro universitet, Datoranvändning-Matlab, 2014.
- [3] Per Jönsson, Mah, Laborationer i Matlab, 2001.
- [4] Gion K Svedberg, föreläsningsanteckningar, 7 nov 2016.

Uppdelning av labbuppgiften

Denna labb går ut på att lära sig de praktiska färdigheterna i att foga Matlab, Arduino och Motor-shielden tillsammans för att läsa in givarevärdena och styra en motor – utan att förstöra hårdvaran!

Vi börjar med att träna upp färdigheterna i Matlabprogrammeringen. Sedan laddar vi ner kommunikationsmjukvaran till Arduino Due så att Matlab och Arduino kan börja kommunicera med varandra. Genom att titta på olika typer av in- och utgångar bygger vi med hjälp av elektroniken upp ett skyddsnät för hårdvaran. Rätt anslutningen mellan Arduino Due och Motorshielden ingår i detta.

När allt är på plats ska ni programmera en liten motorapplikation där man per knapptryck kan byta riktningen av en motor samtidigt som motorströmmen mäts och visas i en graf. Motorn ska sedan stoppas ifall att lasten överstiger ett maximalt värde.

Detta motsvarar en typisk tillämpning av sekventiella styrsystem, som man brukar realisera med hjälp av industriella PLC. Se också föreläsningssanteckningar på its learning om PLC-system, [4]

OBS: Profilen på datorn i labbsalarna som ska startas och användas är: Elektroniklabbet-Windows 7

(Versioner av applikationer som beskrivningen nedan passar till:

- Atmel Studio 6.2
- Matlab 2015b

)

1 Matlabprogrammering

Det finns två korta och enkla introduktioner till Matlab på its learning, se [2] och [3]. Om ni känner att ni behöver en introduktion till Matlab så är det en bra idé att läsa igenom dokumenten samt att göra exemplen.

Det finns mycket hjälp att få, inte minns på youtube. Vad vi kan rekommendera är följande källor:

- i) När man startar Matlab kommer upp en länk "Getting Started" direkt under titeln för "Command Window".
- ii) Om man skriver in "demo" vid Matlabprompten ">>" efter att ha startat upp Matlab...


```
>>demo
```

.. så kommer man åt två videos som ger en översikt över Matlabs möjligheter och användning.
- i) Genom Matlabs hjälp-knapp får man tillgång till dokumentationer, video-clips samt tutorials.

Matlabguide

Det är en bra idé att ha Matlabs Primer 2015b [1] lätt tillgänglig bredvid sig när man programmerar i Matlab. Kapitel 5 om programmering samt kapitel 4 om plotting funktionen är speciellt hjälpsamma!

1.1 Programmering av delta-modulationen från kursen Mobila system

I Lab2 om digital modulering i kursen Mobila system har ni skrivit pseudo-kod eller till och med programmerat delta-moduleringen av en ljudfil. Det är en bra övning att skriva samma

program i Matlab som genererar $dm(k)$ och $y(k)$ för ett ingångssignal $x(k)$ som består av N -antal ($k=1..N$) digitaliserade värden.

Leta fram era program eller pseudo-program och använd dem för att skriva ett Matlab-program. Döp funktionen till "DeltaM". Använd samma parameter eller variabler som från uppgiften, dvs:

- y, x, dm : ska vara array med N element
- k : är en integer index av element i array, ($1 \leq k \leq N$)
- h : upplösning eller tröskelvärde
- y : resulterande signal som resultat av deltamoduleringen (sammansatt av "h"-steg som adderas eller subtraheras som funktion av dm).
- dm : bitstream med "0" och "1" som är signalen som kommuniceras och som kan användas för att räkna ut x .
- $x(k)$: insignalen med värdena som ska delta-moduleras.
- ta en egen ljudfil eller "blomig falukorv" som ligger ute på its learning som $x(k)$

Titta på stommen "DeltaM.m" som finns på its learning för att få ledtrådar om ni behöver. Ni kan använda stommen om ni vill eller göra en hel egen version.

När ni är klara med programmeringen inkl. kommentarer till varje rad och allt fungerar som det ska, ska ni svara på följande frågor:

1.1.1 Klistra in er programkod som fungerar, här:

```
function [ t, x, y, dm ] = DeltaM(filename)
%DeltaM
%  gör en deltamodellering av en ljudfil
%
% ingång: filnamn är textstring av ljudfil inkl. extension, t.ex
% 'test.mp3'
% utgång: dm-array med alla binära styrvärden
%         x-array som motsvarar delta-modulerade signalen

%Variabler:
N=700000; %antal första samples som tas från ljudfil
h=0.01; %minsta kvantiseringseenhet normerad till 1.
yy=[]; Fs=0; y=[]; dm=[]; x=[]; k=0; compare=[]; % variablerdeklaration som egentligen
inte behövs i Matlab

%[yy,Fs] = audioread('falukorv.mp3'); %laddar in en ljudfil och sparar värden i array "yy".
Fs är samplingsfrekvensen av ljudsignalen

%kolla att N inte är större än yy är lång
```

```

if N > length(yy)
    display('N är större än antal samples. N har kortats ner!')
    N=length(yy);
end

y=yy(1:N); %ta första N samples bara
dm=0*y;
x=0*y;
stair=0;
value=0;
compare = 0*y;
t=(1:N); %skapa en tidsvektor

% dm(k) räknas ut som funktion av h(k-1) och y(k-1)
for k=2:(N-1)

    if value < y(k)
        dm(k)=1;
        value = value+h;

    else
        dm(k)= 0;
        value = value+h;

    end

end

end

for k=1:(N-1)

    if dm(k)==1
        stair=stair+h;
        x(k)=stair;

    else
        stair=stair-h;
        x(k)=stair;

    end

end

end

%spela upp resulterande ljudfil x
sound(x,Fs);

yplot = y(400000:400020);
xplot = x(400000:400020);

```

```

dmplot = dm(400000:400020);

plot ((400000:400020),xplot,'b');
hold on
plot ((400000:400020),yplot,'g');
hold on
plot ((400000:400020),dmplot,'black');
title('DeltaModulering')
legend(Utsignal,'Dm','Insignal','Location','soutwest')

end

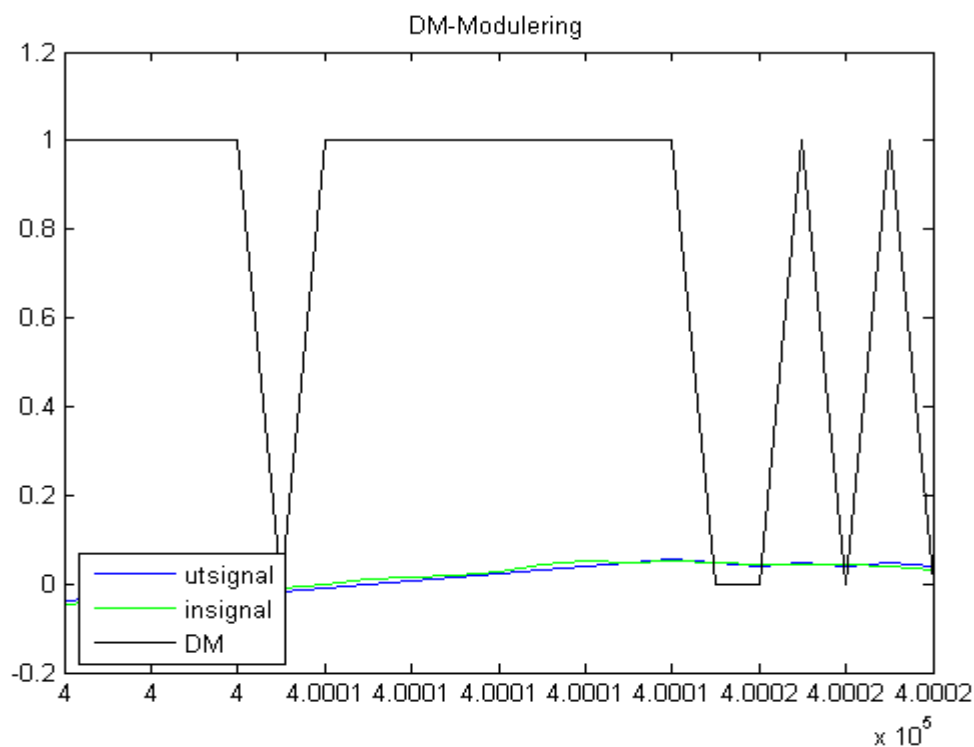
```

1.1.2 Hitta ett bra ställe i signalerna där något händer (dvs original-ljudet inte är konstant) och plotta ca. 20 samples av:

- original-ljudet
- delta-modulerad signal x
- dm

i en och samma plot med olika linjer och färger. Glöm inte att ange vilka kurva som är vilka (=legend). Ange som titel "Delta-modulation".

1.1.3 Spara plottet som en bildfil på hårddisk. Kolla upp Matlabkommandon "print - dtiff". Använd kommandon för att spara bilden och infoga sedan bilden här i word, (OBS- använd inte copy/paste!):



2 Kommunikation mellan Matlab och Arduino Due

2.1 Förberedning av Arduino Due

Innan Arduino och Matlab kan kommunicera med varandra, behöver man ladda ner en liten kommunikations-server applikation till Arduino Due. Det görs genom att genomföra följande steg:

- 1 Anslut Arduino Due till PC:n och starta upp Atmel Studio. Använd mikro-USB anslutning med beteckningen "Programming" närmast extern spänningskontakten.
- 2 Ladda ner och packa upp zip-filen "Matlab-Arduino-master (1).zip" från its learning
- 3 Öppna projektet "DueSrv.atsln" i Atmel Studio
- 4 Ladda ner projektet till Arduino Due, med kommandot "Arduino Due(Debug)" under Tools-menyn.
Finns inte det kommandot i menyn (t.ex. om ni använder era egna datorer) följ instruktionerna som finns som kommentarer i bat-filen "BossacArduinoDue.bat" som finns på its learning i mappen BossacArduinoDue (den går att öppna med en vanlig editor och bygger på <https://github.com/ctbenergy/BossacArduinoDue>). Du måste givetvis uppdatera sökvägarna till var bossac.exe och bat-filen ligger på just din dator. Både bossac-programmet och bat-filen finns i samma katalog på its learning.
- 5 När allt har fungerat som det ska kan ni stänga Atmel Studio

2.2 Förberedning av Matlab

Matlab skapar ett kommunikationsobjekt när den ska kommunicera med Arduino Due. För detta krävs att rätt programvara finns i Matlabs användarkatalog. Genomför följande steg:

- 1 Ta reda på Matlabs aktuella användarkatalog. Byt den till en katalog som ni har kontroll över, dvs som inte finns på labdatorns hårddisk (annars kommer allt att försvinna när ni loggar ut eller om datorn startar om). Bra val är M:/-katalogen eller ett USB-minne.
- 2 Kopiera alla Matlabfiler från its learning till denna arbetskatalog. (Kolla att arduino_com.m, och test_prog.m finns med).
- 3 Kolla upp på vilken com-port Arduino Due kortet är ansluten till PC/MAC. (T.ex. på PC genom "Enhetshanteraren->Portar". Till Enhetshanteraren kommer man genom att trycka "Start->Kontrollpanel->System->Enhetshanteraren").
- 4 Testa kommunikationen genom att skriva följande kommandon i Matlabs kommandofönster (med x= nummer av com-porten från steg 3):
`>> a= arduino_com('COMx')`

Om ni får följande felmeddelande:

```
>> a=arduino_com('COM4')
```

Undefined function or variable 'arduino_com'.

Så betyder det att Matlab inte hittar arduino_com.m-filen i den valda arbetskatalogen. Kopiera dit filen eller ändra arbetskatalogen till rätt ställe.

Det rätta svaret ska vara is stil med:


```
>> a=arduino_com('COM4')  
Attempting connection .....  
Connection successful!
```

```
a =
```

```
Connected to COM4 port
```

```
fx >>
```

2.3 Kommunikation mellan Matlab och Arduino Due

När allt är på plats ska det gå att kommunicera mellan Matlab och Arduino Due.

Kontrollera detta genom att använda er av testprogrammet "test_prog(a)" där variabeln "a" är Matlabs kommunikationsobjektet som ni precis skapade.

Öppna test_prog.m först med en editor för att försöka förstå vad programmet gör innan ni kör det. Som ni ser så används det while-konstruktet "while (1)"-vilket motsvarar en while-slinga med en villkor som alltid är uppfyllt. För att tvinga ett program att sluta exekveras använder man i Matlab tangentkombinationen <Ctrl> C .

Innan ni ansluter komponenter till Arduinon eller testar alla funktioner enligt tabellen 2.3, behöver vi stegvis lägga till skyddande elektronik omkring Arduinon.

Tabell 2.3 Översikt över Matlab instruktioner

Matlab-instruktion	argument	Beskrivning
pinMode(a,pin,state)	a: arduino-objektet pin: pin-nummer (2-13) state:'OUTPUT', 'INPUT'	Initialiserar port med pin-nummer till in- eller utgång.
digitalWrite(a,pin,level)	a: arduino-objektet pin: pin-nummer (2-13) level: 0, 1	Sätter en pinne till 0 eller 1. Måste först deklarerats som utgång
digitalRead(a,pin)	a: arduino-objektet pin: pin-nummer (2-13)	Läser av status (digitalt värdet) av en pin. Bör deklarerats som ingång innan den läses.
analogWrite(a,duty)	a: arduino-objektet Duty: 8-bit PWM signal 0..255	PWM-utgång, låst till DAC1. Använd motorshielden för att styra motorer!!
analogRead(a,pin)	a: arduino-objektet pin: 'A0', 'A1', ('A3')	Läser av en analog pinne. Skall ej deklarerats som ingång innan läsning. AD-upplösning 1024bit

Det vi kan testa just nu är att läsa och skriva digitala in- och utgångar.

2.3.1 Testa ett par av digitala in- och utgångar genom en förbindelse mellan port 3 och 4, där port 3 är en digital utgång och port 4 är en digital ingång. När utgången blir "0" eller "1" ska ingången läsa respektive värden!

- a) Med vilken kommando och hur definieras port 3 till digital utgång och port 4 till digital ingång? Klistra in raderna från Matlabs kommandofönstret här:

```
a.pinMode(3,'OUTPUT')  
a.pinMode(4,'OUTPUT')
```

- b) Hur skrivs ett "0" till utgång 3? Klistra in kommandon från Matlabs fönster här:
`a.digitalWrite(3,0)`
- c) Hur läser man värdet på ingång 4? (Se till att 3 och 4 är förbundna med varandra genom en kabel). Klistra in Matlabraderna här:
`a.digitalRead(4)`
- d) Gör om b) och c) dock med logisk "1"
`a.digitalWrite(3,1)`
`a.digitalRead(4)`

(Tips: Använd tangentbordspilen uppåt för att komma åt och lätt kunna anpassa tidigare kommandon)

3 Signalanpassningar för digitala och analoga in- och utgångar

Att kunna läsa in signalerna vid ingångar och skriva ut värden till utgångar är förutsättning för att kunna styra någonting. För att det ska kunna fungera och inte förstöra hårdvaran krävs en del anpassningar i form av elektronik.

Alla signalerna omkring en Arduino Due behöver begränsas till 3,3V max. Dessutom visar kartan över pinnarna vilka begränsningar av de maximala ingångs- och utgångs-strömmar som finns, (källa: <https://www.arduino.cc/en/Hacking/PinMappingSAM3X>):

Arduino Due pin mapping table

Due Pin Number	SAM3X Pin Name	Mapped Pin Name	Max Output Current (mA)	Max Current Sink (mA)
2	PB25	Digital Pin 2	3	6
3	PC28	Digital Pin 3	15	9
4	connected to both PA29 and PC26	Digital Pin 4	15	9
5	PC25	Digital Pin 5	15	9
6	PC24	Digital Pin 6	15	9
7	PC23	Digital Pin 7	15	9
8	PC22	Digital Pin 8	15	9
9	PC21	Digital Pin 9	15	9
10	connected to both PA28 and PC29	Digital Pin 10	15	9
11	PD7	Digital Pin 11	15	9
12	PD8	Digital Pin 12	15	9
13	PB27	Digital Pin 13 / Amber LED "L"	3	6
54	PA16	Analog In 0	3	6
55	PA24	Analog In 1	3	6
57	PA22	Analog In 3	3	6
67	PB16	DAC1	3	6

Kartan över pinmappningen.

3.1 Anpassa och skydda Arduino Dues digitala utgångar

Anta att man vill tända och släcka en ljusdiod med en digital utgång. Som pinmappningen ovan visar så har de flesta digitala pinnar upp till 15 mA utgångsström. Det borde räcka för att få en ljusdiod att tända. Problemet är bara att inte alla LED klarar 3,3V spänning som Arduino Due har på sina utgångar. Här behövs ett motstånd i serie som skyddar LED.

3.1.1 Undersök ljusdioden som ni har lånat. Kolla försiktigt med en power-box och en voltmeter vid vilken spänning dioden börjar lysa:

a) vilken är den minimala spänningen:

≈1,8V

b) vilken kan anses som den "normala" spänningen, dvs där ni anser ljusdioden lyser normalt:

2V

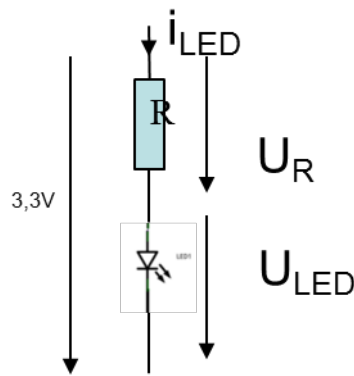
c) Vilken ström mäter ni med spänningen från a) ?:

≈15mA

d) Vilken ström mäter ni med spänningen från b)?:

≈27mA

Bestäm motståndet som ni behöver sätta i serie med LED genom att använda er av följande sammanhang:



$$U_R + U_{LED} = 3,3V$$

$$U_R = i_{LED} * R$$

3.1.2 Räkna ut värdet R för motståndet:

- Hur stor är i_{LED} ? 20mA
- Hur stor är U_{LED} ? 2V
- Hur stor är U_R ? $3,3 - 2 = 1,3$
- Hur stor är R ? $1,3 / 20 * 10^{-3} = 65 \Omega$
- Vilken värd för R hittar ni i skåpet som passar bäst?
68Ω

Anslut nu lysdioden med motståndet R i serien mellan pin 5 och GND (ground). OBS diodens polaritet!

Sätt pin 5 med Matlab till digital output (använd pinMode).

3.1.3 Sätt utgången av pin 5 till "1" med digitalWrite. Om inte LED brinner, kolla dess polaritet! Mät med voltmeteren de olika spänningarna:

- Hur stor är U_{LED} ? 2,1V
- Hur stor är U_R ? 0,72V
- Hur stor är U vid pin 5? 2,83V
- Hur bra stämmer värdena överens med beräkningarna?
Stämmer ganska bra $2,1 + 0,72 = 2,82V$ jämfört med det uppmätta värdet 2,83

3.1.4 Lämna LED och motståndet R mellan pin 5 och ground. Utför sedan testprogrammet test_prog.m.

- Vad förväntar ni er att programmet ska göra?
Led lampan ska blinka
- Hur exekverar man programmet, dvs vad skriver ni in Matlabs Kommandofönster?
Test_prog(a)
- Blev resultatet som förväntat? Om inte, varför inte?
Ja

3.2 Arduino Dues digitala ingångar (Repetition av pull-down och pull-up)

So länge man håller ingångsspänningen under 3,3 V klarar de digitala ingångarna sig. Ansluter man de till ground (GND) kommer de att läsa "0", ansluter man de till 3,3V kommer de att läsa "1". Problemet kan uppstå när de inte är anslutna till någonting alls eller till trådar som hänger lösa. Då är dem inte definierade och värden som läsas in är slumpartade. För att förhindra detta kan man använda sig av 10kOhm pull-down eller pull-up motstånd, om man till exempel vill använda sig av enkla knappar. Problemet med enkla knappar är att beroende

om de är anslutna till ground eller 3,3V är de bara definierade när man trycker ner dem, men inte om de är öppna. Det är därför att det behövs pull-down eller pull-up motstånd.

Testa de digitala ingångarna 4-9 genom att:

- Sätta alla till digitala ingångar med pinMode
- Läs in deras värde med digitalRead
- Mäta deras spänningspotential med en voltmeter

3.2.1 Fyll i tabellen med era resultat:

Pin	Digitala värdet	Spänningen V
4	1	3,25
5	1	3,25
6	1	3,25
7	1	3,25
8	1	≈ 0
9	1	≈ 0
10	1	3,25
11	1	3,25
12	1	3,25

(pin 10-12 ska inte initialiseras med pinMode! Det ska illustrera skillnaden mellan initialiserade och icke-initialiserade digitala pinnar).

3.2.2 Anta att ni vill ansluta en vanlig enkel knapp så att det blir 3,3V vid ingången när den trycks ner och annars är det 0V.

a) Hur ser kretsen ut med en pull-down motstånd? Rita kretsen och förklara:

I en pulldown krets så vill man sänka spänningen till 0V när enheten är aktivt ansluten.

b) Hur ser kretsen ut med en pull-up motstånd? Rita kretsen och förklara:

I en pullup krets så höjer den spänningen till dess spännings source level när inkopplad komponenterna är inaktiva

c) Vilken av de både kretsarna väljer ni och varför?

Vi väljer pulldown eftersom vi vill få ner spänningen till 0V när vi kopplar komponenter på arduino kortet.

d) Vilken av de både kretsarna finns inbyggd i Arduino Due? Hur ser man det?

Arduio Due har en inbyggd pullup resistor eftersom vi fick en 1:a när vi läste från dem olika pinnarna.

3.2.3 Bygg upp kretsen som ni har bestämt er för under c) ovan med ett motstånd på 10kOhm vid port 4 och enkel knapp som ni ansluter till +3,3V. OBS: var försiktig att hitta 3,3V anslutningen på Arduino Due och inte 5V som ligger precis bredvid!

- a) Klistra in resultatet ni får när ni läser in värdet med digitalRead och en knapp som inte är tryckt och sedan när den är tryckt:

Ingen knapp intryckt = 0V, digitalt värde=0

Knapp intryckt = 3,3V, digitalt värde=1

- b) Mät med voltmeteren spänningen vid ingången till port 4 när knappen inte är tryckt:

När knapp INTE intryckt $\approx 0V$
Och när knappen är tryckt: 3,3V.

3.3 Arduino Dues analoga utgångar

Som översikten över pinnarna visar, så har pin "DAC1" som används som analog utgång en maximal utgångsström av bara 3mA. Det är rund 100 gånger för litet för att driva den typen av motor som vi använder i denna lab! Även spänningen med bara 3,3V är antagligen för lite.

Använd power-boxen och multimeteren för att reda ut spänningen och strömmen för motorn som ni har lånat.

3.3.1 Börja med 0V och höj sedan spänningen tills motorn börjar snurra.

- a) Vilken minimal spänning mäter ni över motorn?
 $\approx 150mV$
- b) Vilken motorström mäter ni vid denna spänning?
 $\approx 90mA$
- c) Ök spänningen försiktigt tills ni känner att det borde vara motorns normala spänning. Vad mäter ni för spänning?
 $\approx 3V$
- d) Vad mäter ni för ström?
335mA

I de allra flesta fall behövs kopplingen via en motor-shield för att kunna styra en motor.

OBS: Stoppa inte in motor-shielden som sköld på arduino due! Har den istället bredvid på plattan, så som ni har lånat den!

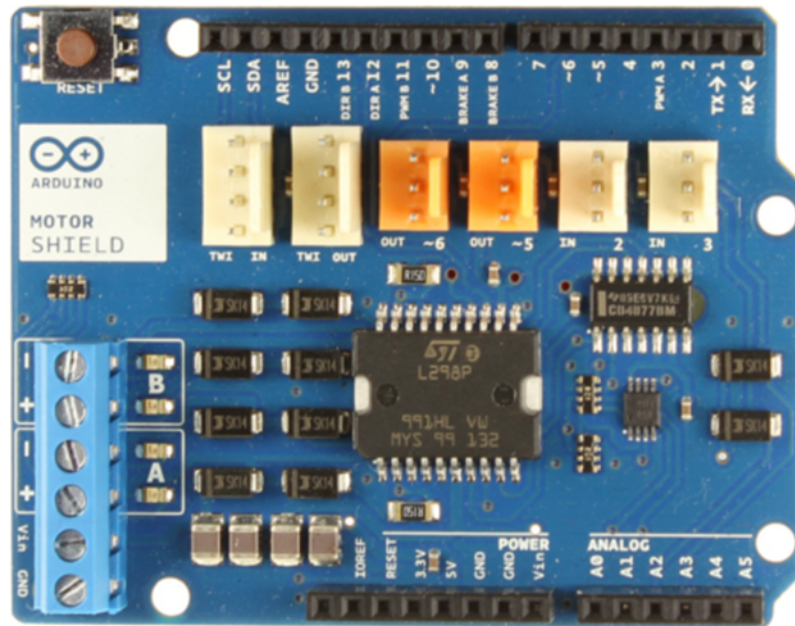
Börja med att titta närmare på skölden och de viktigaste pinnarna:

Input and Output

This shield has two separate channels, called A and B, that each use 4 of the Arduino pins to drive or sense the motor. In total there are 8 pins in use on this shield. You can use each channel separately to drive two DC motors or combine them to drive one bipolar stepper motor.

The shield's pins, divided by channel are shown in the table below:

Function	pins per Ch. A	pins per Ch. B
<i>Direction</i>	D12	D13
<i>PWM</i>	D3	D11
<i>Brake</i>	D9	D8
<i>Current Sensing</i>	A0	A1



Använd kanal A för att driva motorn.

3.3.2 Vilka av de 4 pinnar som beskrivs ovan är

a) ingångar till motor-shielden?

A och B är kanalerna för motorer med 4 ingångar var

A har ingångarna D12,D9,D3

B har ingångarna D13,D11,D8

b) Utgångar från motor-shielden?

A har utgången A0

B har utgången A1

3.3.3 Hur ska man ansluta Arduino Due till motor-shielden? Svara på frågorna utan att genomföra anslutningarna än!!

- a) a) vilka arduino pinnar ska anslutas till sköldens ingångar?
DAC1→D3 PWM, 3→9 break, 2→12 direction
- b) Vilka arduino pinnar ska anslutas till sköldens utgångar? Vad bör beaktas?
A0 skall skyddas och leda till A0
- c) Vilka andra arduino pinnar ska anslutas till skölden?
5v→5v, GND→GND,
- d) Vad mer behöver anslutas till motor-skölden för att kunna driva motorn?
(Jämför också med föreläsningssanteckningarna)
En extern motorspänning (Powerbox), samt en motor som kopplas till kanal A

Vi ska nu testa om vi kan få sätta på och stänga av motorn med arduino tillsammans med motor-shielden. Följ noggrant följande steg i rätt ordning:

1. Lämna kvar installationen av lampan och knappen som ni har gjort tidigare, de kommer att användas senare igen.
2. Anslut motorn till kanal A på motor-shield (MS), det spelar ingen roll vad som är + eller –
3. Anslut externa spänningskälla (power-box) till MS, minus ska till ground, plus till Vin! Ställ in samma spänning som ni mätt vid motorn.
4. Förbind port DAC1 vid Arduino Due (AD) med pwmA-pinnen på MS
5. Förbind GND på MS med GND på AD
6. Förbind +5V från AD till +5V på MS

3.3.4 Hur får man nu igång motorn?

- a) Vilket kommando ska man ange i Matlab?
a.analogWrite(255)
- b) Hur, med vilket kommando, stänger man av motorn?
a.analogWrite(0) eller a.digitalWrite(3,1)

3.3.5 a) Hur byter man riktningen som motorn snurrar på? Ange Matlabkommandot.

a.digitalWrite(2,0) eller a.digitalWrite(2,1)

3.3.6 Skriv nu ett litet Matlabprogram som gör att motorn sätts på och sedan byter riktning varje gång man trycker på en knapp!

Klistra in koden av er Matlabprogram här:

```
dir=0;
a.pinMode(3,'OUTPUT');
a.digitalWrite(3,0);

pause(0.5);

a.pinMode(2,'OUTPUT');
a.digitalWrite(2,0);

pause(0.5);

a.analogWrite(254);
while(1)
btn = a.digitalRead(6);
pause(0.2);

if btn == 1

    if dir==0
        dir=1;
    else
        dir=0;

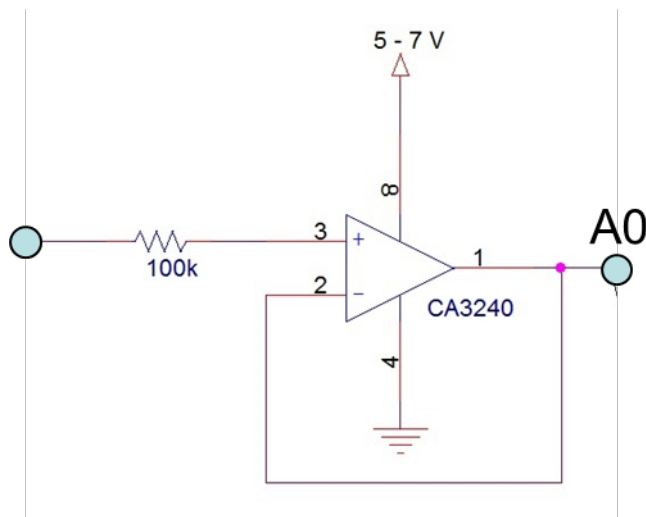
    end
    disp(dir);
    pause(0.8);
    a.digitalWrite(2,dir);
end
end
```

Tips: med <Ctrl><C> kan man stoppa ett Matlabprogram.

3.4 Arduino Dues analoga ingångar A0 och A1

Vi behöver en analog ingång (A0) i denna lab. För att förhindra att den kan gå sönder behöver den skyddas med den elektroniska kretsen som ni gjorde i en tidigare lab. Där använde ni en operationsförstärkare (CA3240) för att hålla ingångsspänningen mellan -0.3V till +3,6V.

Figuren nedan visar kretsen som kan byggas för att skydda ingången A0:



Figur: Kretsen från tidigare labb som kan skydda en mikroprocessoringång

Operationsförstärkarens 5-7V spänning borde kunna tas från Arduinos 5V utgång tillsammans med motor-shielden. Om det är för lite så ta den andra spänningskällan på power-boxen.

Vi vill mäta motorströmmen som motor-shielden räknar ut med vår analog ingång A0. Problemet är bara att motor-shielden arbetar med +5V och Arduino Due med max. +3,3V. Vi behöver vara säkert att motorshieldens spänning på A0 inte överstiger 3,3V. Annars finns det risk att Arduino Dues A0-ingång kan förstöras! Om A0 kan överstiga 3,3V behöver ni bygga ihop skyddskretsen enligt figuren ovan.

3.4.1 Mät spänningen på pinnen A0 på motorshielden när ni kör motorn. I vilken intervall ligger mätvärden om ni ökar lasten på motorn, t.ex. genom att hålla rotorn mellan tummen och pekfinger?

Spänningen ligger runt 1,35 V när vi ökar lasten.

3.4.2

a) Vad drar ni för slutsatser från era mätningar angående motorshildens spänning på A0? Att vi inte kommer att behöva ett skydd på A0, eftersom spänningen inte överstiger 3,3V

b) Vad hittar ni för information om motorshildens spänning på A0? Citera relevant information och anger referensen.

”On each channel will be a voltage proportional to the measured current, which can be read as a normal analog input, through the function `analogRead()` on the analog input A0 and A1. For your convenience it is calibrated to be 3.3V when the channel is delivering its maximum possible current, that is 2A.”

Referens: <https://www.arduino.cc/en/Main/ArduinoMotorShieldR3>

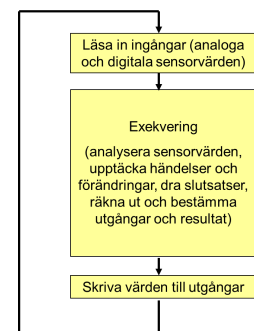
Med era svar från frågorna ovan kan ni lätt avgöra hur kretsen från figuren ovan ska förbindas med motor-shielden.

4 Styrteknik

Grattis! Om ni har kommit så lång så har ni nu ett fungerande styrsystem som kan ta in och skriva ut digitala och analoga värden och som kan programmeras med Matlab.

Ni ska nu skriva ett styrprogram enligt PLC-tekniken där man cykliskt läser in sensorvärden, sedan analyserar värden och räknar ut resultat som man till slut skriver ut till utgångarna. Detta repeteras cykliskt tills programmet stängs av. Kom ihåg att det är den del från teorin enligt föreläsningen (se [4]), som vi vill tillämpa i denna uppgift! Denna struktur krävs därför i era program om de ska godkännas, även om det finns enklare och smidigare andra lösningar.

Cyklisk exekvering



Till förfögen finns två olika ”process”: en leksaks-kran som kan lyfta en liten last samt en byggsats-hiss som kan dra upp och ner en slags korg. Antal process är starkt begränsat, så ni behöver först utveckla ert program med motorn som ni har och sedan, när allt fungerar, byta motorn mot processen.

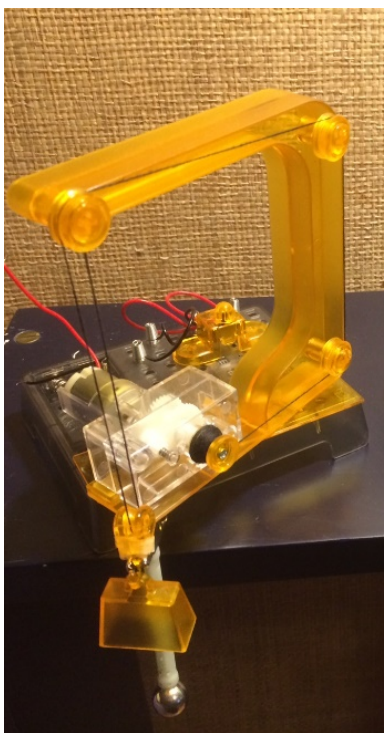


Fig. Kort av de två olika process som man kan välja emellan för att utföra slutuppgiften på

Syftet med slutuppgiften är att föra alla olika delar ihop och kolla om man har förstått det som krävs för att uppnå och dokumentera styrningen av en process.

Slutuppgift

Hitta lösningar till följande problem:

1. Styr riktningen av motorn med en knapp. Varje gång knappen trycks ska riktningen bytas.
2. En LED ska lysa när hissen eller kranen åker uppåt och vara släkt när motor stannar eller åker i andra riktningen.
3. Motorströmmen ska mätas och mätvärden ska sparas i en Matlabvariabel som sedan kan plottas. Minst 200 mätpunkter ska få plats i variabeln.
4. Motorströmmen ska mätas med minst två olika lastar med olika vikt. Varje mätning ska sparas i egna variabler.
5. Variablerna med olika motorströmmen ska plottas i samma graf med egna kurvsymboler, kurvbeskriftningar, och titel.
6. Variablerna med olika motorströmmen ska dessutom överföras direkt till ett excel-fil. (Kolla upp Matlab-kommandon med namn "xlswrite" för att ta reda på hur man gör).
7. Lägg till en funktion i motorstyrningen som övervakar motorströmmen och som direkt bromsar motorn om strömmen överstiger en viss gränsvärd som ligger lite över värden för när motorn drar den tyngre lasten.

Dokumentation av er lösning:

4.1 Klistra in Matlab-programmet och ange vilka tre delar i programmet som motsvarar den cykliska delarna vid exekvering av PLC-styrteknik:

```
function elevatorLed(a)
pause on
dir=1;

a.pinMode(3,'OUTPUT');
a.digitalWrite(3,0);

a.pinMode(2,'OUTPUT');
a.digitalWrite(2,dir);

a.pinMode(7,'OUTPUT');
a.digitalWrite(7,0);

a.analogWrite(0);

startMeasure = 0;

pause(0.4);

goToCase = 0;
valuesLight = zeros(1,300);
valuesHeavy= zeros(301,600);
i = 1;
j = 1;
filename = 'data.xlsx';
```

```

while(1)

%SAMLA MÄTVÄRDEN FÖR LÄTT VIKT
if (startMeasure == 1)
    i=i+1;

    %GÖR OM MÄTVÄRDEN TILL mA
    valuesLight(i)=(a.analogRead('A0')/1024)*2000;

    %STANNA OM MÄTVÄRDEN FÖR MOTORSTRÖM ÖVERSTIGER 175 mA
    if valuesLight(i) >= 175
        a.digitalWrite(3,1);
        disp('Too heavy! Breaking!');
    end
end

%SAMLA MÄTVÄRDEN FÖR TUNG VIKT
if (startMeasure == 2)
    j=j+1;

    %GÖR OM MÄTVÄRDEN TILL mA
    valuesHeavy(j)=(a.analogRead('A0')/1024)*2000;

    %STANNA OM MÄTVÄRDEN FÖR MOTORSTRÖM ÖVERSTIGER 175 mA
    if valuesHeavy(j) >= 175
        a.digitalWrite(3,1);
        disp('Too heavy! Breaking!');
    end
end

%NOTIFIERA OM ATT VÄRDENA FÖR LÄTT VIKT ÄR SAMLADE
if i == 300
    disp('Done collecting the first 300 values');
end

%KOLLA OM ALLA VÄRDEN ÄR SAMLADE OCH RITA UT DEM
if(i>=300 && j>=300)

    plot(valuesLight(1:300));
    hold on
    plot(valuesHeavy(1:300));
    title('Motorström för olika vikter')
    xlabel('Arrayindex')
    ylabel('mA')
    legend('Lätt vikt', 'Tung vikt', 'Location', 'southwest')
    a.analogWrite(0);
    xlswrite(filename,valuesLight(1:300),'A1:KN1');
    xlswrite(filename,valuesHeavy(1:300),'A2:KN2');
    break;

end

```

```

%Läs in knappingången
btn = a.digitalRead(6);

if btn == 1

    %ÅK NER
    if goToCase == 2
        a.analogWrite(200);
        goToCase = 3;

        %STANNA OCH STÄLL IN RIKTNING
    elseif goToCase == 3
        % a.digitalWrite(7,0);
        a.analogWrite(0);
        dir=1;
        goToCase = 0;

        %ÅK UPP
    elseif goToCase == 0
        a.analogWrite(200);
        a.digitalWrite(7,1);

        startMeasure = startMeasure + 1;

        goToCase = 1;

        %STANNA OCH STÄLL IN RIKTNING
    elseif goToCase == 1
        dir=0;
        a.digitalWrite(7,0);
        a.analogWrite(0);
        goToCase = 2;
    end

    pause(0.2);
    a.digitalWrite(2,dir);

end

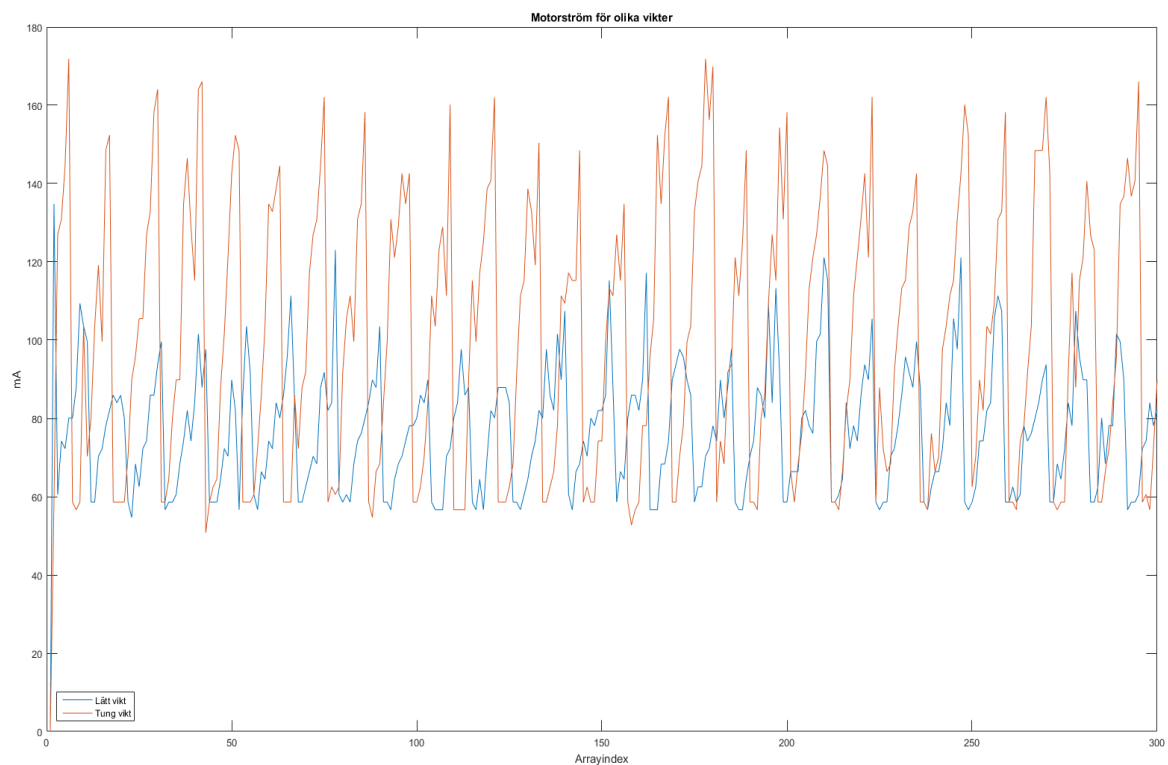
end

end

```

1. I whileloopen så läser vi värden från ingångarna ifall en knapp trycks ner
2. Efter kör vi diverse exekveringar för att kolla ifall villkoren uppfylls
3. Skulle det uppfyllas så skriver vi ut till utgångarna för motorn och lampan beroende på villkoret

4.2 Klistra in er plot som tiff-bild, här:



4.3 Kopiera/infoga delar av excel-filen som visar att ni har lyckats att överföra Matlabs variabler:

103,515 625	97,65 625	105,46 875	97,65 625	99,609 375	93,75	58,593 75	58,593 75	58,5937 5	68,359 375
99,6093 75	58,59 375	60,546 875	58,59 375	62,5	70,31 25	68,359 375	80,078 125	123,046 875	95,703 125

4.4 Spela in en video av ert styrsystem som visar hur hissen eller kranen fungerar med ert program och hur motorn stannar när lasten blir för tungt. Ladda upp videon på its learning eller kopiera en länk till videon här:

-----Test av system med mätvärden-----
https://drive.google.com/open?id=0B1QNzgklhAX_cU1IZDVLUmJWcUU

-----Test av system break funktion-----
https://drive.google.com/open?id=0B1QNzgklhAX_MnJpajZXbk5FazA

5 Utvärdering och reflektion över det egna lärandet

5.1 Vad tycker du/ni var lärorik med uppgiften? (Minst 5 meningar och minst 75 ord!)

Vi fick använda oss av matlab för att programmera, både som programspråk och verktyg för att plotta ut olika grafer. Vi har lärt oss om hur motorshielden fungerar tillsammans med arduino due kortet. Vi har lärt oss hur man kan använda matlab på olika sätt för att generera olika excelfiler med värden från matlab samt kunna plotta ut dessa senare. Vi fick även lära oss vilka risker som finns gällande höga inspänningar på dem olika komponenterna och hur dessa kan ta skada ifall man matar in för mycket spänning.

5.2 På vilket sätt har ni fördjupat er i något nytt? Vad kände ni från tidigare och på vilket sätt har ni lärt er något nytt utifrån det ni redan kunde? (Minst 5 meningar och minst 75 ord!)

Vi har fördjupat oss en hel del inom matlab som programspråk och matematiskt verktyg. Vi har lärt oss om hur motorshielden fungerar tillsammans med arduino, tidigare har vi styrt motorer utan motorshield direkt på arduino vilket var något nytt för oss och roligt. Vi kunde lite om hur ut och ingångar fungerar och fick nu använda det mer tillsammans med matlab som fungerade lite annorlunda från vanliga arduinos programverktyg. Vi har även fördjupat oss om hur vi kan använda oss av Fluke som mätinstrument.

5.3 Vad var det svåraste med uppgiften? (Minst 5 meningar och minst 75 ord!)

Hur mycket tid totalt har ni lagt ner på att lösa uppgiften och hur mycket av denna tid har ni lagt på det som ni anser var det svåraste?

Vi har haft ganska mycket småfel som tagit ganska lång tid. Exempelvis så hade vi en trasig kabel som tog oss ganska lång tid att upptäcka efter ett antal undersökningar fram och tillbaka. Det har varit ganska mycket strul med datorerna på skolan som gjort att matlab inte fungerat bra. Det har varit lite problematiskt att hitta en hiss som även är fungerande. Vid något tillfälle stängdes datorn av och tog lite tid att starta upp datorn igen. Matlab har ibland inte fungerat plötsligt och man var tvungen att starta om för att det skulle fungera med tillsammans med programmet.

5.4 Synpunkter, förslag, kommentarer? (Minst 5 meningar och minst 75 ord!)

Labbarna har varit ganska roliga men ibland fastnade man och visste inte riktigt vad som var felet och fick undersöka en del. Det hade underlättat ifall skoldatorerna fungerade bättre och det fanns föremål som kranar till alla så att man inte behövde vänta tills man fick en. Det hade varit roligare ifall man kunde bygga ut systemet lite med olika sensorer för att få det att likna mer ett PLC system exempelvis genom att lägga till någon sensor som känner av och därefter göra något.