ECEN 5053-001

Lab 5:

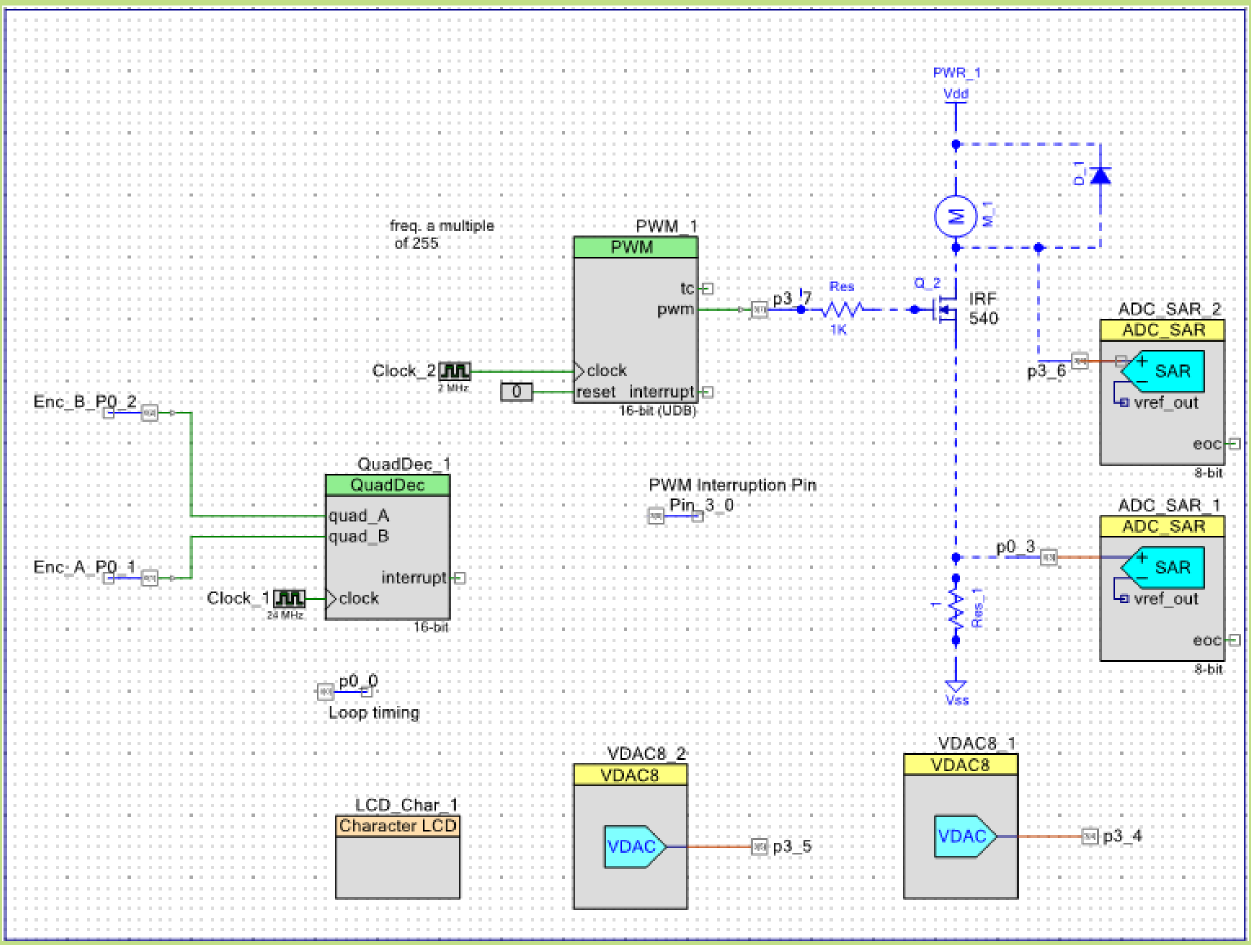
<u>DC brush motor Speed and Current Control</u>

10-19-17

Monish Nene

Nagraj Siddeshwar

<u>Goal:</u> Control the speed of dc brush motor using feedback from encoder and back emf. Also control the current of same.



Equipment and Parts Required: Cypress PSoC CY8CKIT-050B development board, power supply, digital oscilloscope, dc brush motor with encoder and gearbox (Pololu item 2822, supplied), fast recovery power diode.

Step 1- (speed control using encoder)

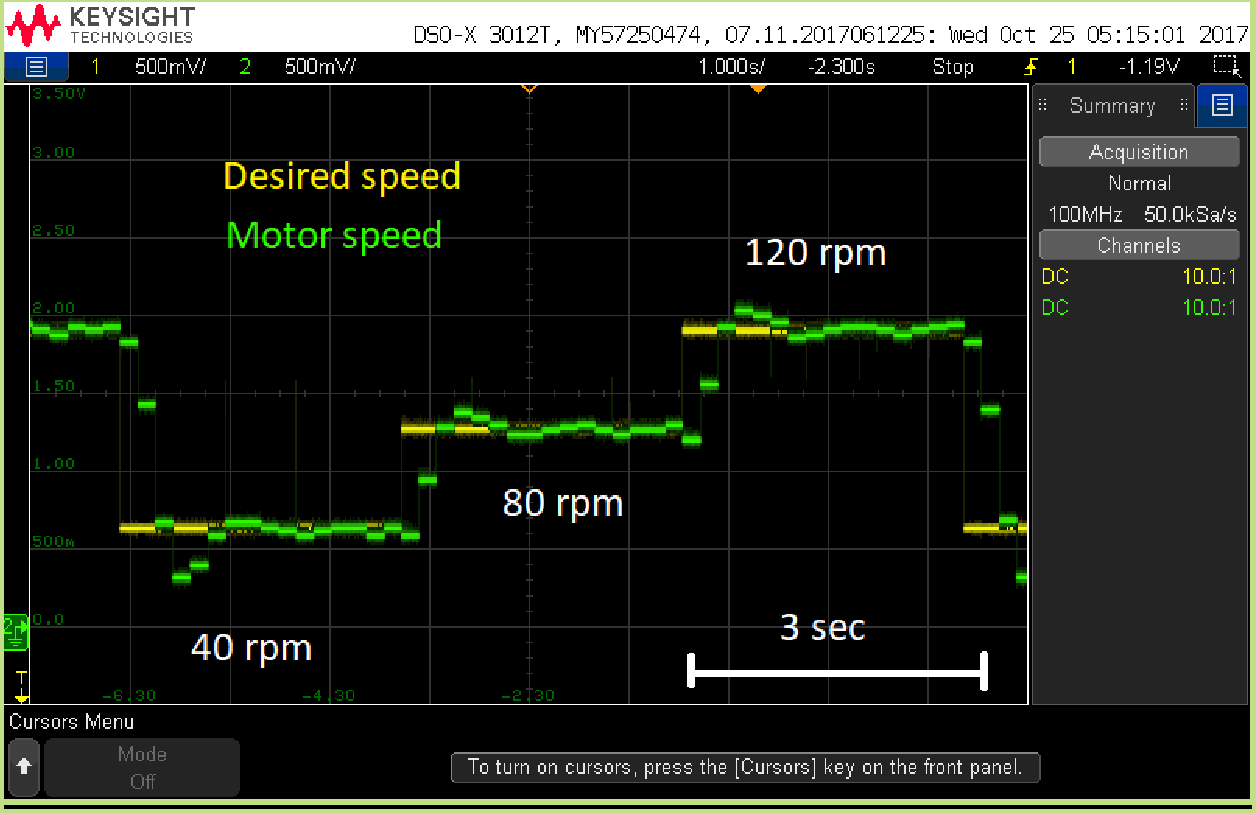
Using feedback from the encoder (established in the previous lab), *use it to control the motor speed* by automatically changing duty cycle of the FET according to whatever equation you choose (as long as it works).

To demonstrate speed control, the desired or “reference” motor speed is cycled between 3 different values. First, it should be set first at 40 rpm and held for 3 seconds, then switch to 80 rpm and hold that for 3 seconds. Finally go to 120 for 3 seconds. Repeat this cycle continuously. Good control is indicated by a quick speed change and accurate tracking of the reference speed.

The LCD should display the desired speed in rpm on the first row (on the left side) followed by the motor PWM value (on the right side). On the second row, show the speed measured by the encoder directly below the desired speed for comparison.

Required demo 1: show this to your instructor or TA for credit.

Scope shot 1: ch 1 = desired speed from a first DAC, ch 2= measured speed from encoder from a second DAC.

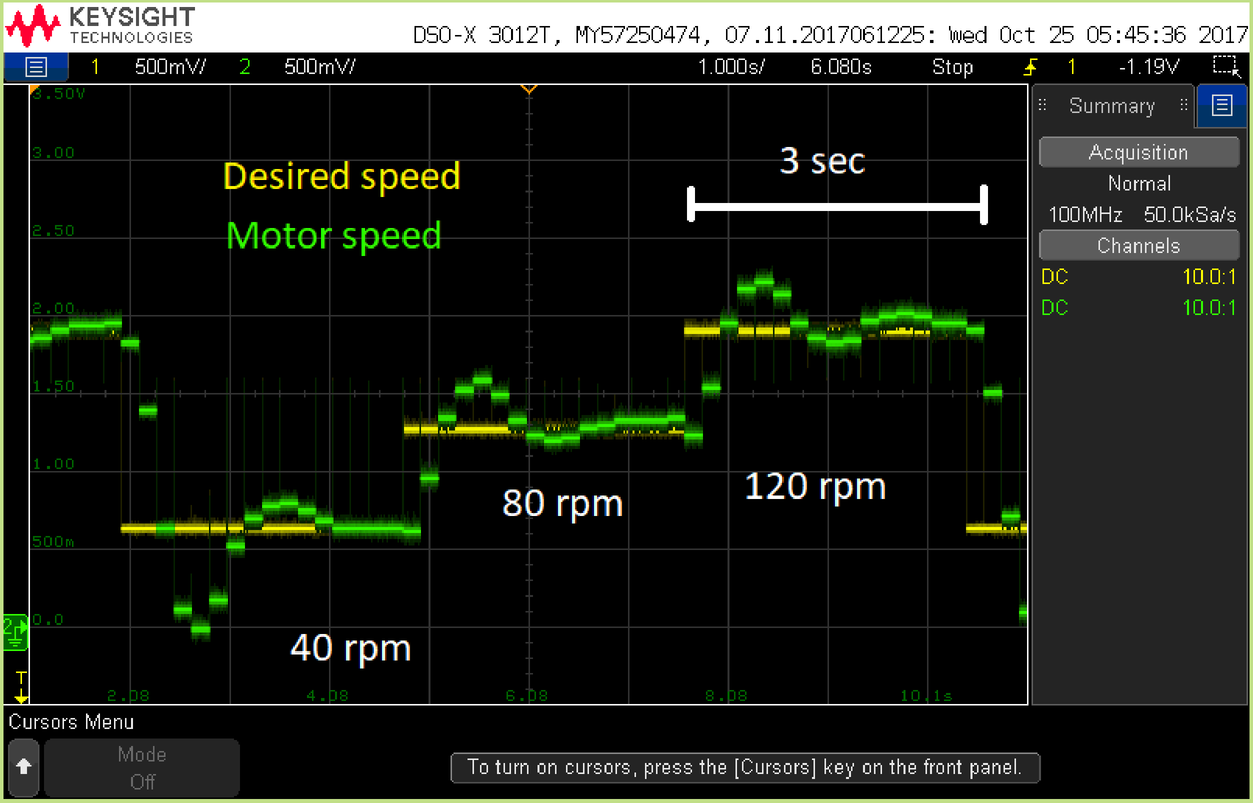


<u>Step 2</u>- (speed control using back EMF)

Repeat step 1 except this time use feedback from back EMF.

<u>Required demo 2:</u> show this to your instructor or TA for credit.

<u>Scope shot 2:</u> ch 1 = desired speed from a DAC, ch 2= measured speed from back EMF from another DAC.

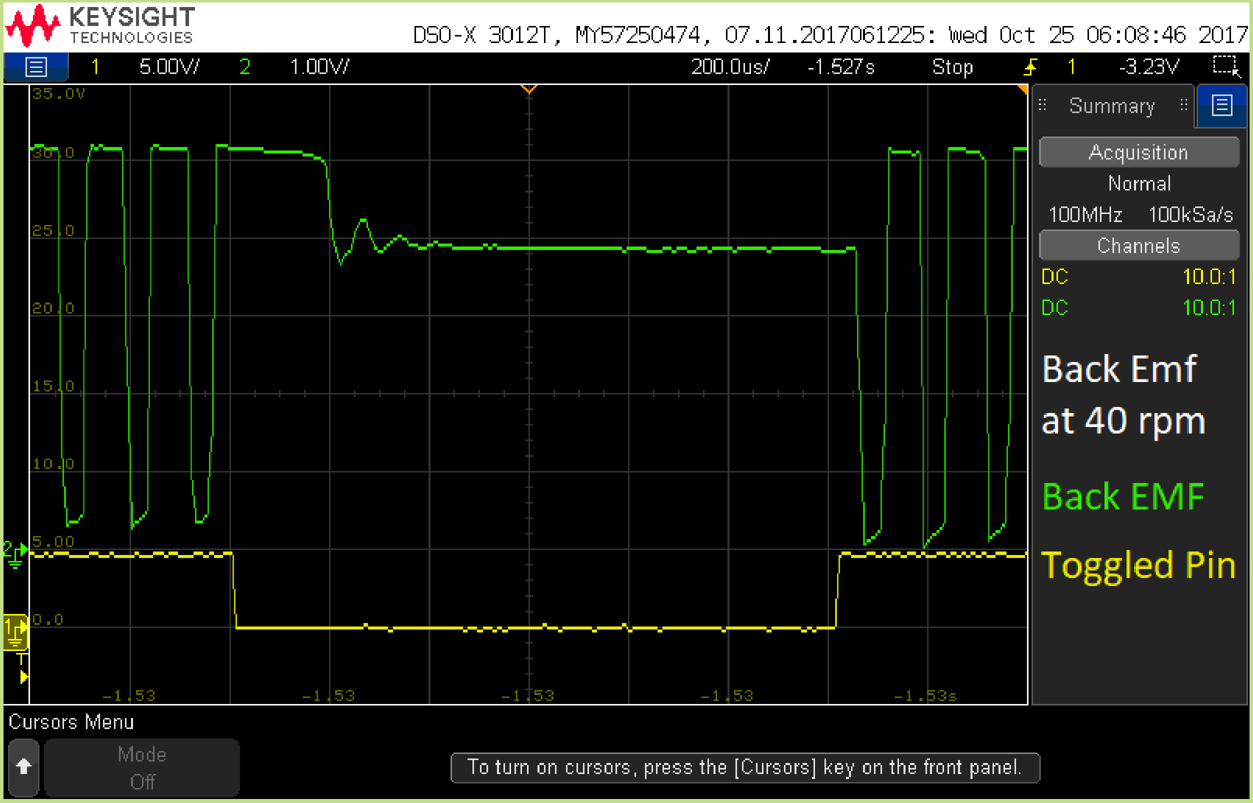


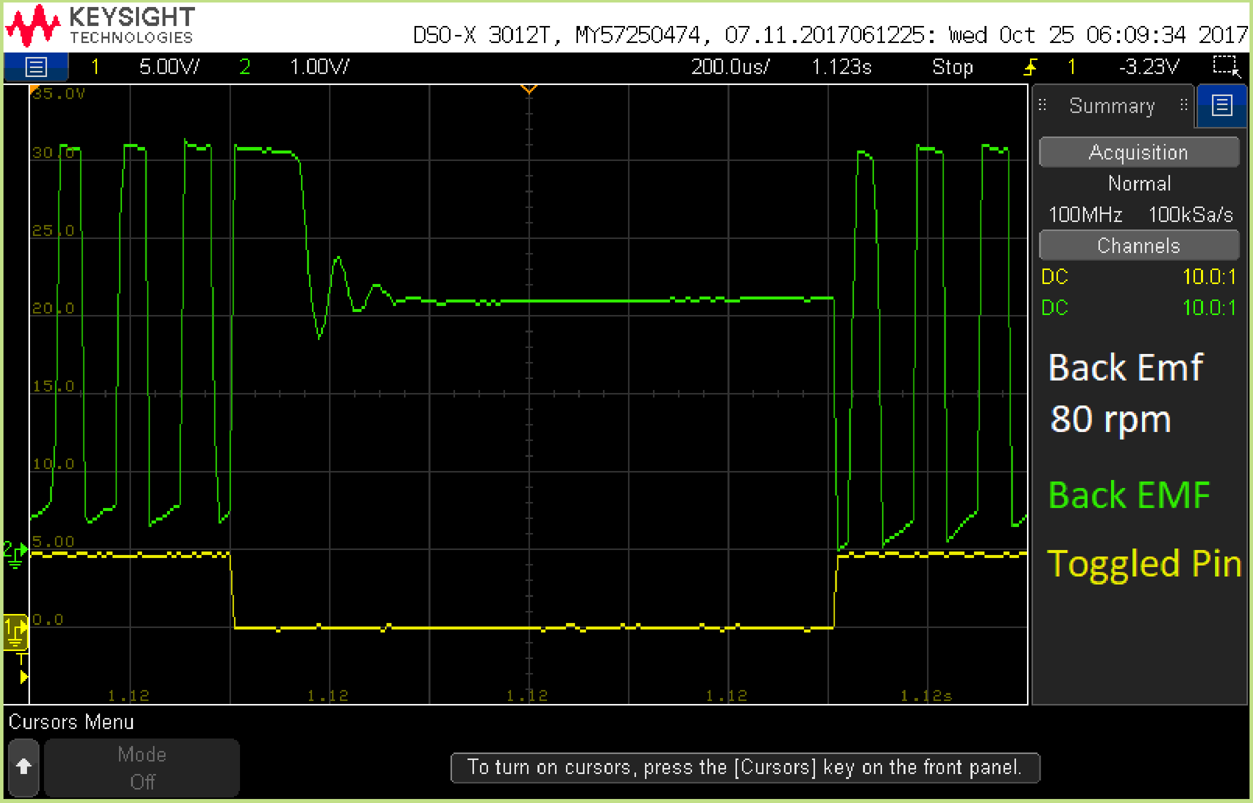
<u>Scope Shots 3-5:</u> Flip a pin at the start of when you turn off the motor voltage (/pwm) and turn it off when you finish measuring the back emf

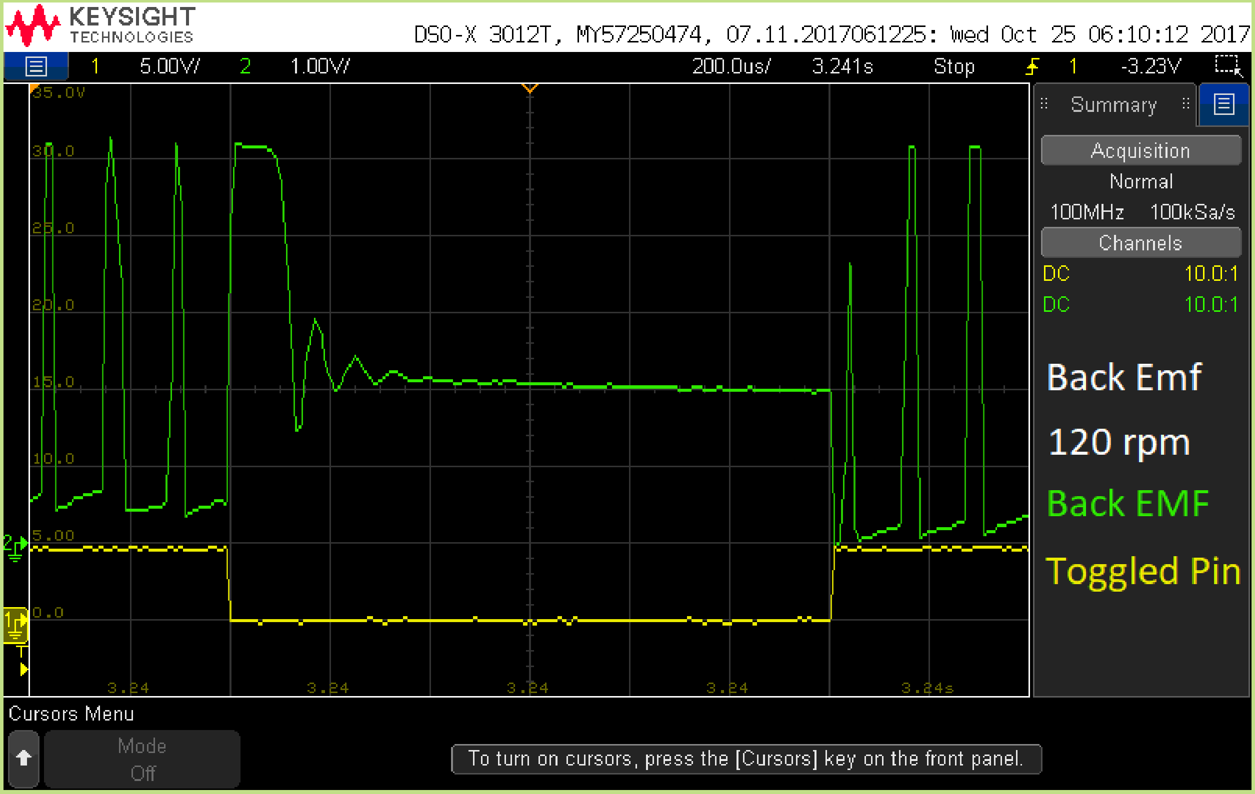
Ch1: flipped Pin Ch2: motor voltage

This is a good indication of what voltage you're measuring from the back emf.

Do this at 3 different speed levels.


KEYSIGHT TECHNOLOGIES
DSO-X 3012T, MY57250474, 07.11.2017061225: Wed Oct 25 06:08:46 2017
1 5.00V/ 2 1.00V/ 200.0us/ -1.527s Stop 1 -3.23V
Summary
Acquisition
Normal
100MHz 100kSa/s
Channels
DC 10.0:1
DC 10.0:1
Back Emf
at 40 rpm
Back EMF
Toggled Pin
Cursors Menu
Mode
Off
To turn on cursors, press the [Cursors] key on the front panel.


KEYSIGHT TECHNOLOGIES
DSO-X 3012T, MY57250474, 07.11.2017061225: Wed Oct 25 06:09:34 2017
1 5.00V/ 2 1.00V/ 200.0us/ 1.123s Stop 1 -3.23V
Summary
Acquisition
Normal
100MHz 100kSa/s
Channels
DC 10.0:1
DC 10.0:1
Back Emf
80 rpm
Back EMF
Toggled Pin
Cursors Menu
Mode
Off
To turn on cursors, press the [Cursors] key on the front panel.



<u>Question 1:</u> What calculation did you use to set the FET duty cycle?

Answer:

```
for(;;)
{
PWM_1_WriteCompare(inertia);
//3 sec mode change
if(mode==1)
      {speedref=40;}
if(mode==2)
      {speedref=80;}
if(mode==3)
      {speedref=120;}

//Feedback section

//error = speedref - rpm; //for encoder rpm

error = speedref – emfrpm; /for backemf rpm

inertia = inertia + error; //value to be written to the PWM

if(inertia>255|inertia<0)//handling math overflow
      {inertia=100;}

//delay 3 sec        if (loop % 1600000 == 0)
        {
        loop=0;
        if(mode==3)
```

```
            {mode=1;}
            else
            {mode++;}
            }
}
```

<u>Step 3:</u> (Motor current control with shaft locked).

<u>**Required in report:**</u> show your circuit and code. Explain in words how it works.

Answer:

1) The ISR block is attached to the interrupt of PWM and the interrupt is given when the PWM goes from high to low.
2) At this time a delay of 20us is given to overcome noise and the ADC measures the voltage across resistance.
3) The Voltage and converted to current and then compared with current current_ref.
4) The error is added to the Value of PWM input (inertia). Only a part of the error (0.2 times) is added to avoid overshoot.
5) If the value of inertia is out of bonds then the previous value is used. (This will limit the operational region of the motor current)
6) An additional increment or decrement is done in inertia just to get a quicker response.(This is done in main loop)
7) A loop count is kept to change the value of current_ref every 3 seconds.

ISR Loop

```
 CY_ISR(isr_2_Interrupt)
{   PWM_1_STATUS;
    p0_0_Write(!p0_0_Read());
    CyDelayUs(20);
    source_voltage = ADC_SAR_1_GetResult8();
    current = (source_voltage*2048)/(255.0 * 2.7);
    //delay 3 sec
    ierror = current_ref - current;
    inertia = inertia + 0.2 * ierror;
    VDAC8_1_SetValue(current);
    VDAC8_2_SetValue(current_ref);
    PWM_1_WriteCompare(inertia);
    if(inertia > 200)//handling math overflow
        {inertia=prev_inertia;}
    else if (inertia < 50)
        {inertia=prev_inertia;}
     prev_inertia=inertia;
}
```

Main Loop

```
    PWM_1_Stop();
    if(current_ref > current)
    {inertia ++;}
    else
    {inertia--;}
    PWM_1_Start();
    }
    if (loop % 100000 == 0)
    {LCD_Char_1_Position(1,0);
            LCD_Char_1_PrintString("current:        ");
            LCD_Char_1_Position(1,8);
            LCD_Char_1_PrintU32Number(current);
            LCD_Char_1_Position(0,0);
            LCD_Char_1_PrintString("ref:     ");
            LCD_Char_1_Position(0,4);
            LCD_Char_1_PrintU32Number(current_ref);
            LCD_Char_1_Position(0,8);
            LCD_Char_1_PrintString("duty:   ");
            LCD_Char_1_Position(0,13);
            LCD_Char_1_PrintU32Number((100*inertia/255.0));
            /*LCD_Char_1_Position(1,11);
            LCD_Char_1_PrintString("e:   ");
            LCD_Char_1_Position(1,13);
            LCD_Char_1_PrintU32Number((100*inertia/255.0));*/
     }
     if (loop % 450000 == 0)
     {
     loop=0;
     if(current_ref==100)
     {current_ref=200;}
      else
        {current_ref=100;}
     }
}
```

Required demo 3 : show this to your instructor or TA for credit.

Question 2: What problems do you encounter if the current sense resistor value you choose is too large (how large is too large)?

Answer: We can't get enough current flowing through the motor if the resistor is too large. Too large is 2v (minimum voltage motor needs to work on)/200ma (max current required) – internal resistance of motor = 500 ohms – internal resistance of motor. Any value above 10

ohms can be considered too large as it is an unnecessary load and increases power consumption.

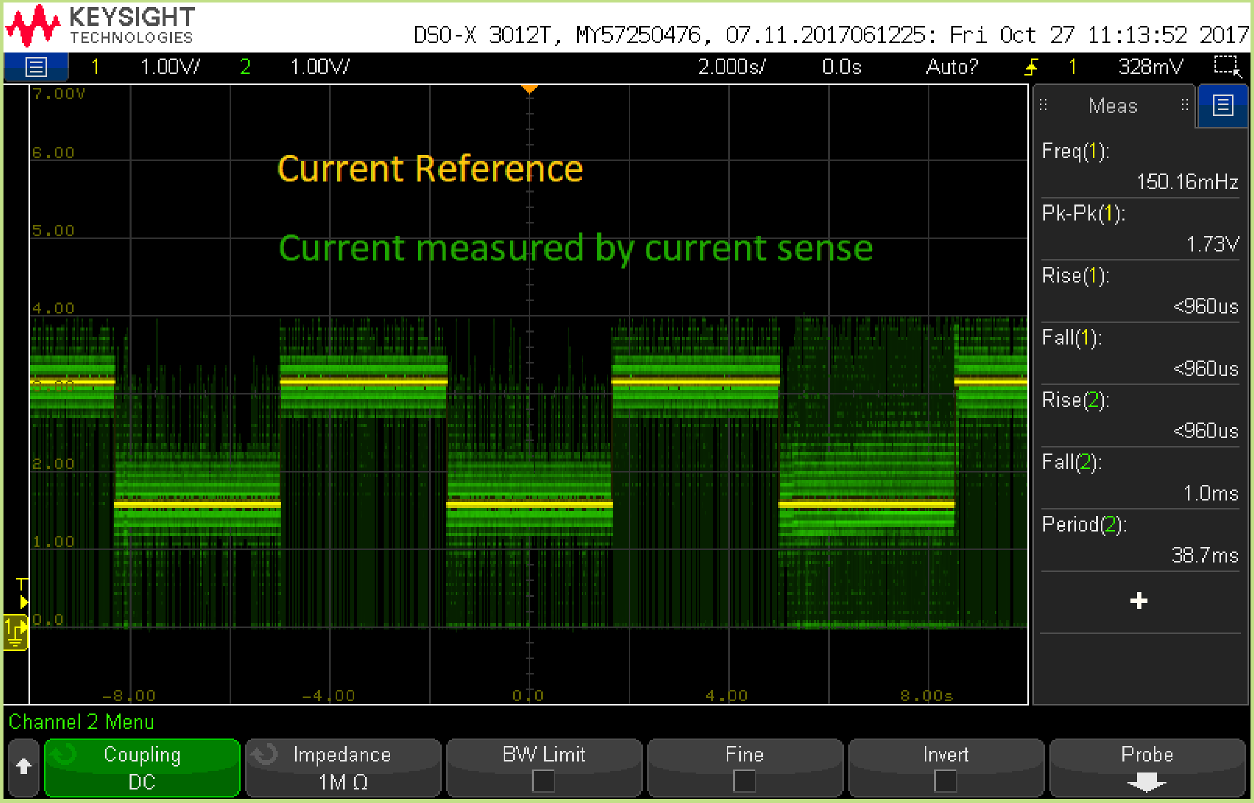
Question 3: What problems do you encounter if the current sense resistor value you choose is too small (how small is too small)?

Answer: If the value is too small then the voltage across the resistor is too small and hard to measure. We will need to add a PGA before giving voltage value to ADC. Any value below 1 ohm is too small.

Scope shot 6: ch 1 = desired current value (from the DAC) as it changes from low to high (vertical scale should match that used for ch 2).

ch 2= current as measured by the current sense resistor.

Answer:



Question 4: How large is the current ripple?

Answer: The current ripple is approximately 100mV/2.7 ohm = 37.ma for me as I was using a 2.7 ohm resistance.

Question 5: What happens to the current control if the motor shaft is allowed to turn freely?

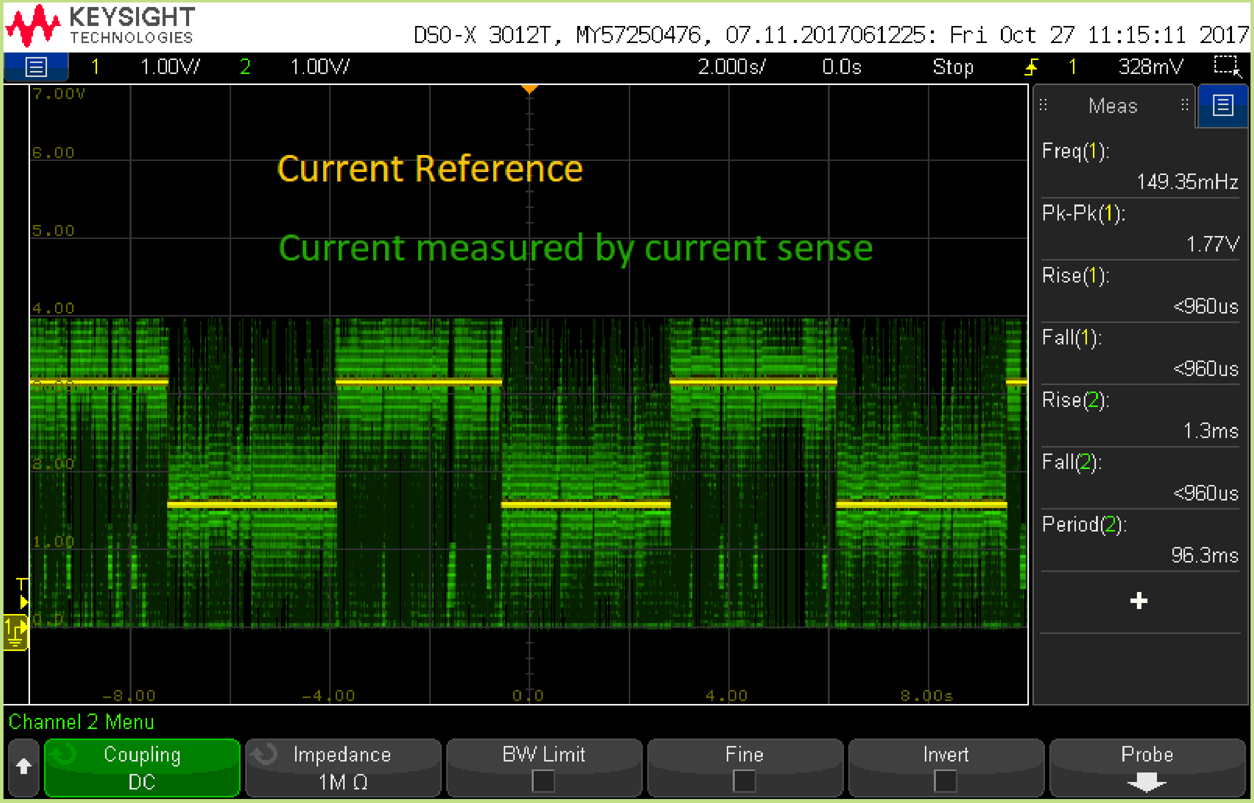
Answer: If the motor shaft is allowed to move freely the current is harder to control because of BACK EMF.

Question 6: What happens if the motor shaft is allowed to turn but you add a little drag to it to slow it down.

Answer: There are disturbances in the measured current due to noise but the value doesn't change. The motor adjusts the PWM to maintain constant torque when little load of finger is applied.

Scope shot 4: Show the current changing from 100 to 200 mA in the middle of the trace while the motor is turning slowly (add a little drag with your finger).

Answer:



<u>**Question 7:**</u> what is the current level when the FET first turns on?

Answer: The current level is nearly 180mA when the FET first turns on.

############################## End ############################

0.