Electrical, Computer & Energy Engineering
UNIVERSITY OF COLORADO BOULDER

ESM Lab Grading Report

Course Number: 5
Module: 5: Lab 2 on Close Loop Motor Control
Lab Report Date: 06/12/2018

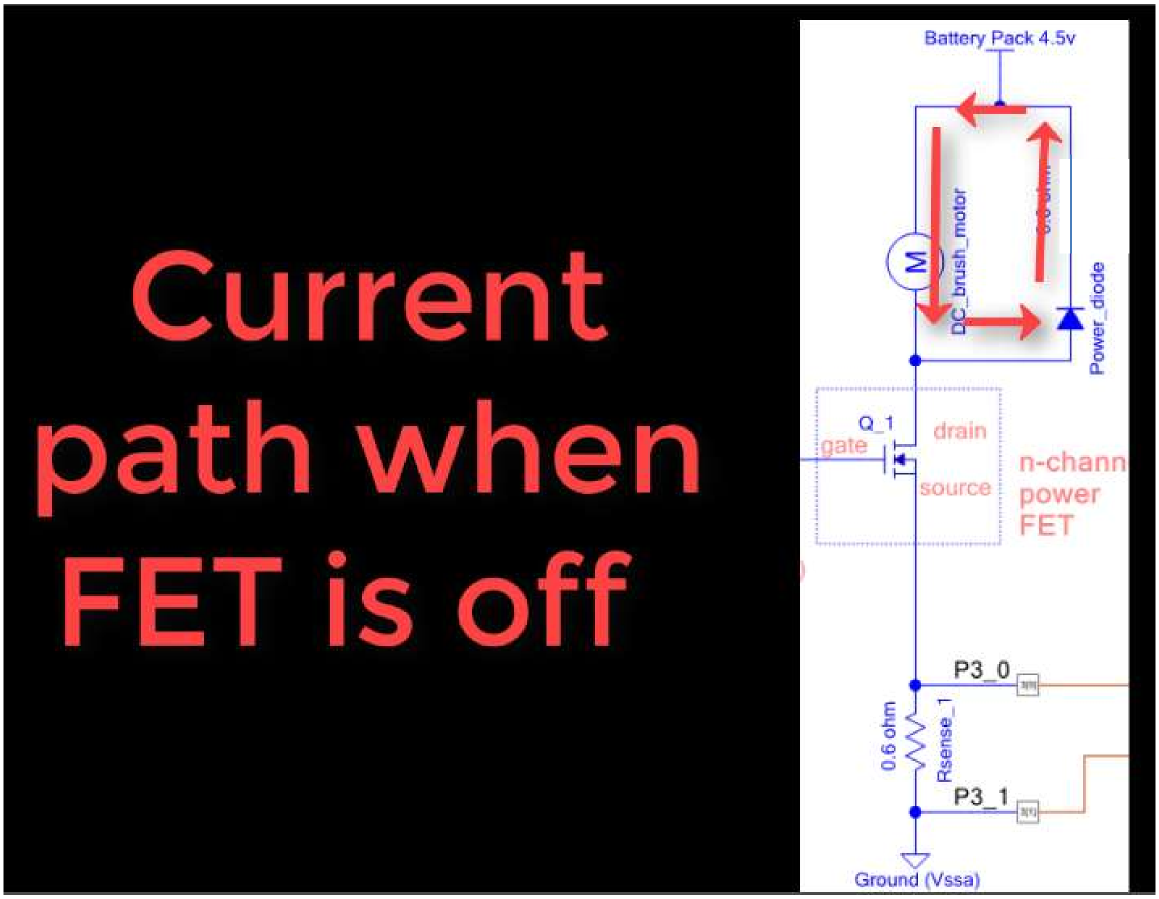
Student Name(s): **Rushi James Macwan and Poorn Mehta**

Each Section of this Lab Report Counts for 20 points. Points will be allocated as follows:
20 points: section fully meets requirements of this rubric
15 points: section mostly meets requirements of this rubric
10 points: section meets roughly half the requirements of this rubric
5 points: section does not meet requirements, but shows a weak attempt
0 points: section blank

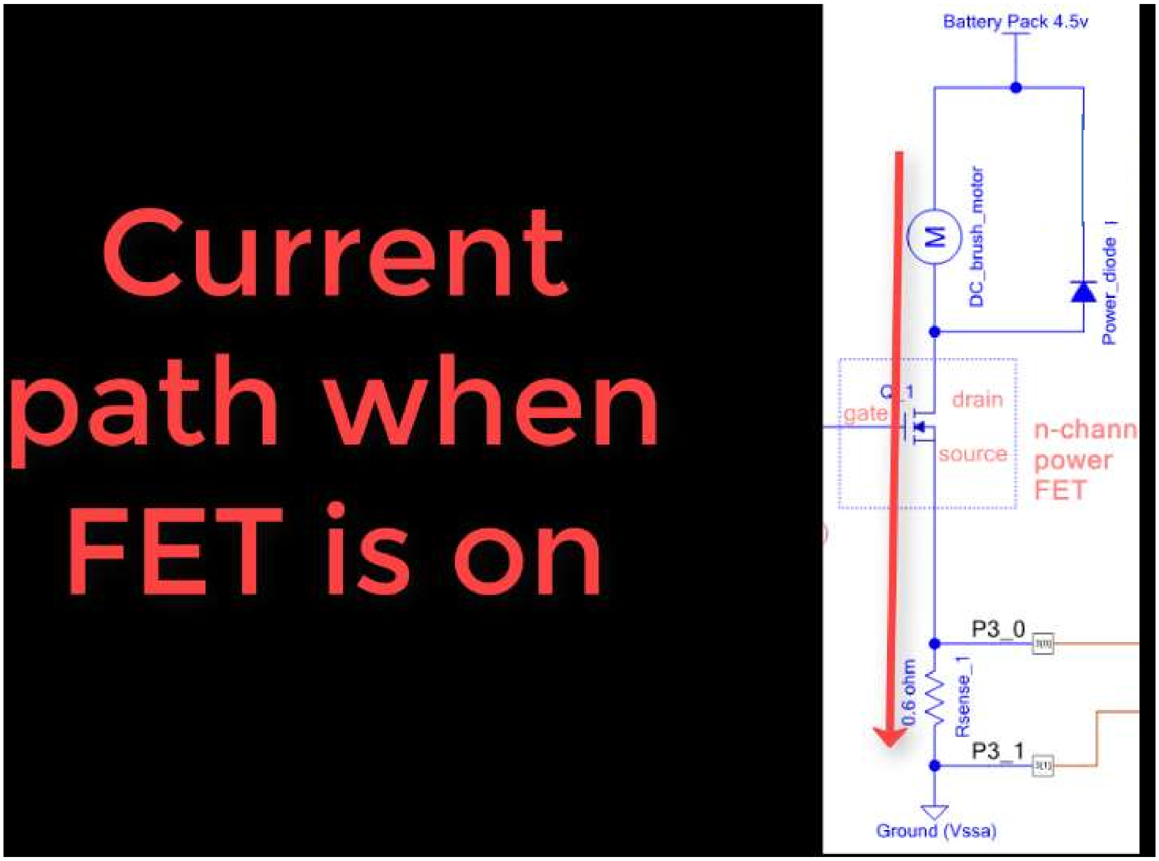
Goal: The purpose of this lab is to learn how to create a circuit that executes close loop motor control of a small DC motor. We need only use a small DC motor, power transistor, current sense resistor, and "flyback" diode to do this. All other components we will use are found inside the PSOC system. For this lab we will use exactly the same external components that we used for "Lab 2 on Motor Voltage and Current Measurement". We will add internal components in our PSOC system to control the current in the motor.

Background: Current is most easily sensed using a resistor, but there are other means to do this. For instance, you can measure it via the magnetic field that is induced when current flows through a wire. Recall that the back EMF is proportional to the speed of the motor, with the constant of proportionality K_e measured in volts per thousand RPM. If you stall the motor, by holding the output shaft with a clamp, you will have zero back EMF. Nevertheless, you will still get significant current flowing through the motor. At that point the PSOC circuit can control the current to any value within the range of the available voltage divided by the resistance of the motor wiring. The current will be controlled through the 0.6 ohm resistor. Make sure you do not use a wire wound resistor, as it will have inductance as well as resistance.

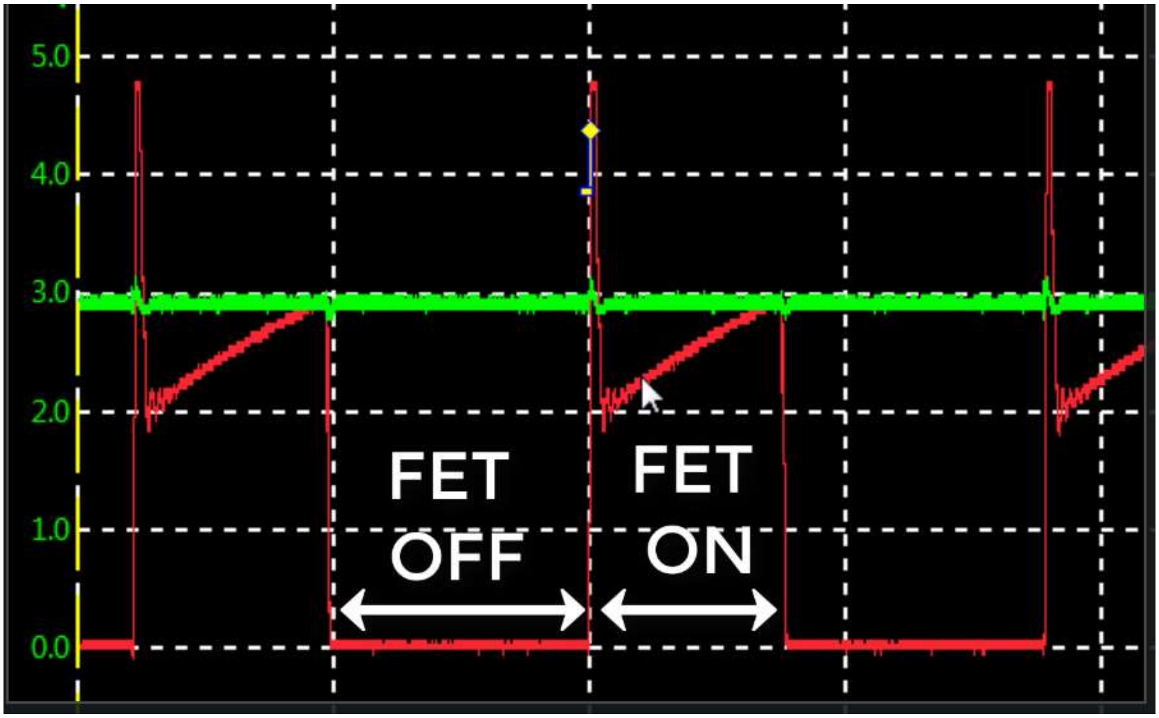
The power diode in the circuit is necessary. When the FET turns off, the current in the motor needs a place to flow. In an inductor, V = L di/dt. When the FET turns off the rate of change of current is abrupt, and the voltage will rise. The power diode provides a path for the current to flow. When this happens the current flows counterclockwise through the power diode, per the first diagram below. However, it will not flow into the 0.6 ohm current sensor resistor. Hence, you will get a measurement of current through the current sense resistor only when the FET is turned on, per the second diagram shown below.



The current will decay during the time that the FET is off, so we will shut it off for a fixed amount of time. You create a desired current in the motor using a DAC. The voltage level is compared via hardware to the output of the DAC, and the FET is turned on and off in response to the result of that comparison. You use your PSoC software to set the level of the DAC.



The motor current is shown in the red trace of this scope trace below. When the FET is on, the motor current rises, limited by the inductance of the motor. The green trace represents the desired current in the motor, as set in the PSoC software. When the motor reaches the desired current, the FET is turned off and the current in the motor quickly decays. Please watch the video for full details on how this works.



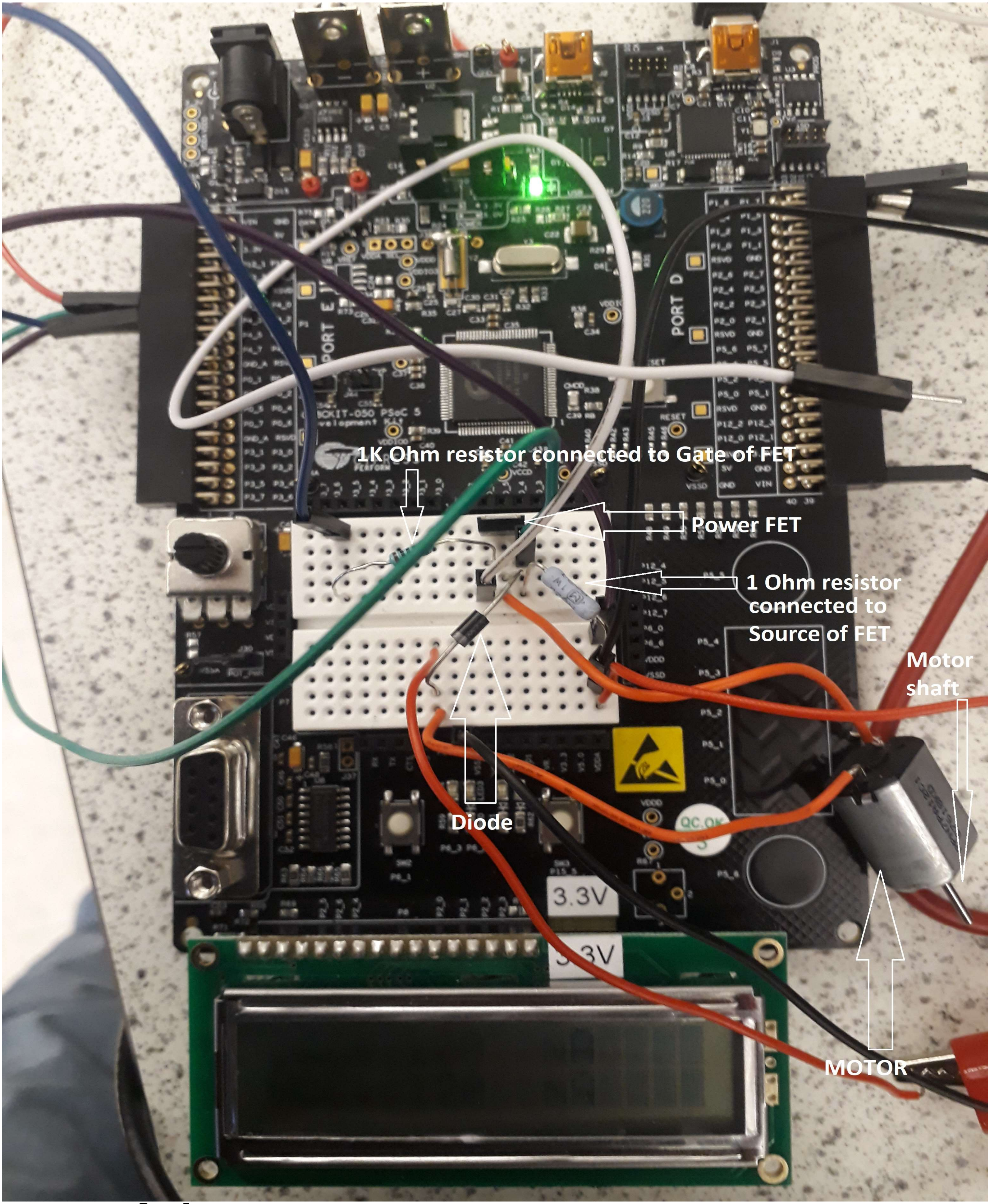
(A) Functional demonstration of your circuit to our TA. In this exercise, you schedule an appointment with your TA to show that your hardware functions as designed. For the Closed Loop C Motor Control lab, this will involve the following steps:

1. Show that all hardware is in place, and that your PSoC software can control the motor current.
2. Show the scope traces of your current flow, along with the desired current setting from the DAC.

If you are an on-campus student, then show your circuit to one of our TA's during office hours.

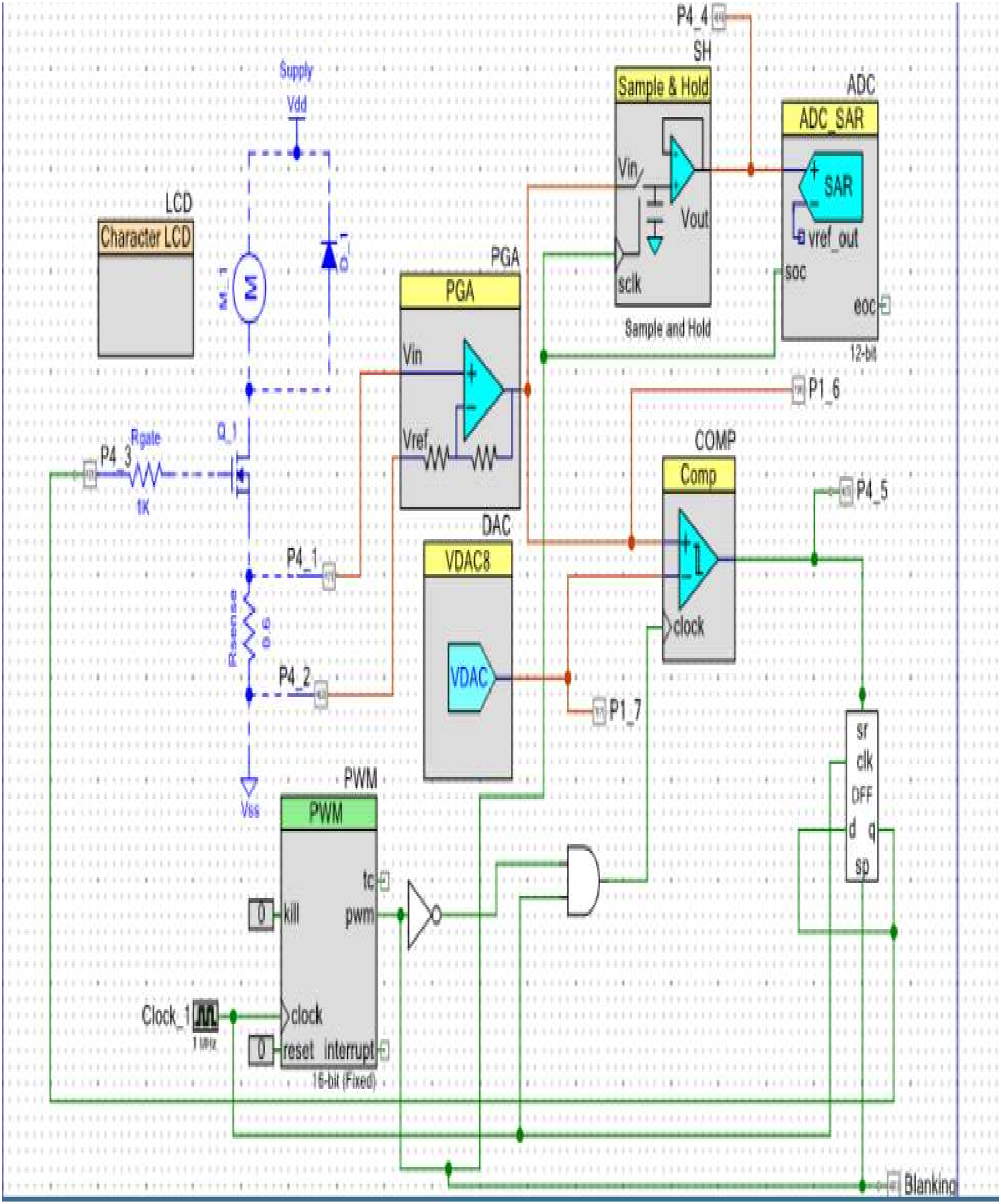
If you are a distance learning student, make an online appointment with your TA to demonstrate your work via Zoom meeting or other Web-based meeting tool. You can use the camera on your laptop PC or suitable plug-in webcam (Logitech etc.) to demonstrate a working circuit.

(B) Place photos here of your hardware setup, including PSoC board, connections to Oscilloscope or nScope, wiring, LCD Display, components, etc.
Label all components.


1K Ohm resistor connected to Gate of FET
Power FET
1 Ohm resistor connected to Source of FET
Motor shaft
Diode
MOTOR
PORT D
PORT E
3.3V

(C) Place complete PSoC schematic here. This schematic must include internal components from the PSoC board (amplifier, ADC, etc.), as well as external components (power diode, motor, n-channel power FET, etc.).

Sample schematic is shown here. (Make sure to delete the sample and place your own schematic here).



P4_4
SH
Sample & Hold
Vin
Vout
sclk
Sample and Hold
ADC
ADC_SAR
SAR
vref_out
soc
eoc
12-bit
Supply
Vdd
LCD
Character LCD
M_1
D_1
PGA
PGA
Vin
Vref
P1_6
COMP
Comp
clock
Rgate
P4_3
1K
Q_1
P4_5
DAC
VDAC8
P4_1
Rsense
P4_2
VDAC
P1_7
Vss
PWM
PWM
tc
kill
pwm
sr
clk
DFF
d
q
sp
Clock_1
clock
reset
interrupt
16-bit (Fixed)
Blanking

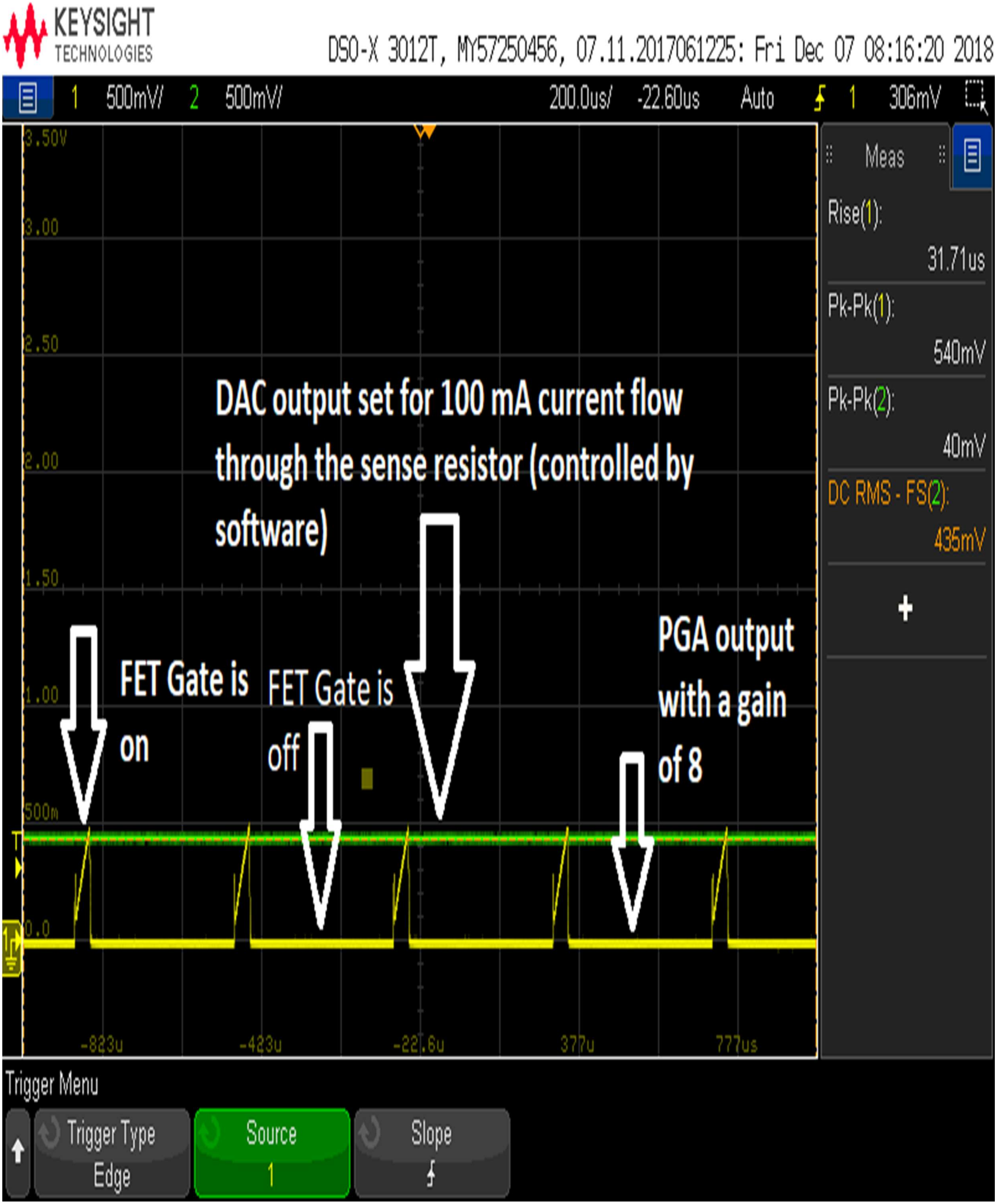
(D) Complete PSoC software. This software must include calls to all internal functions, appropriate comments, and functional code that you included. We will not grade you on the exact syntax and structure, as there are numerous ways to structure the code and still provide the temperature measurement function. Instead, we will grade you on the completeness of the code relative to using the appropriate PSoC functions to gather the necessary data.

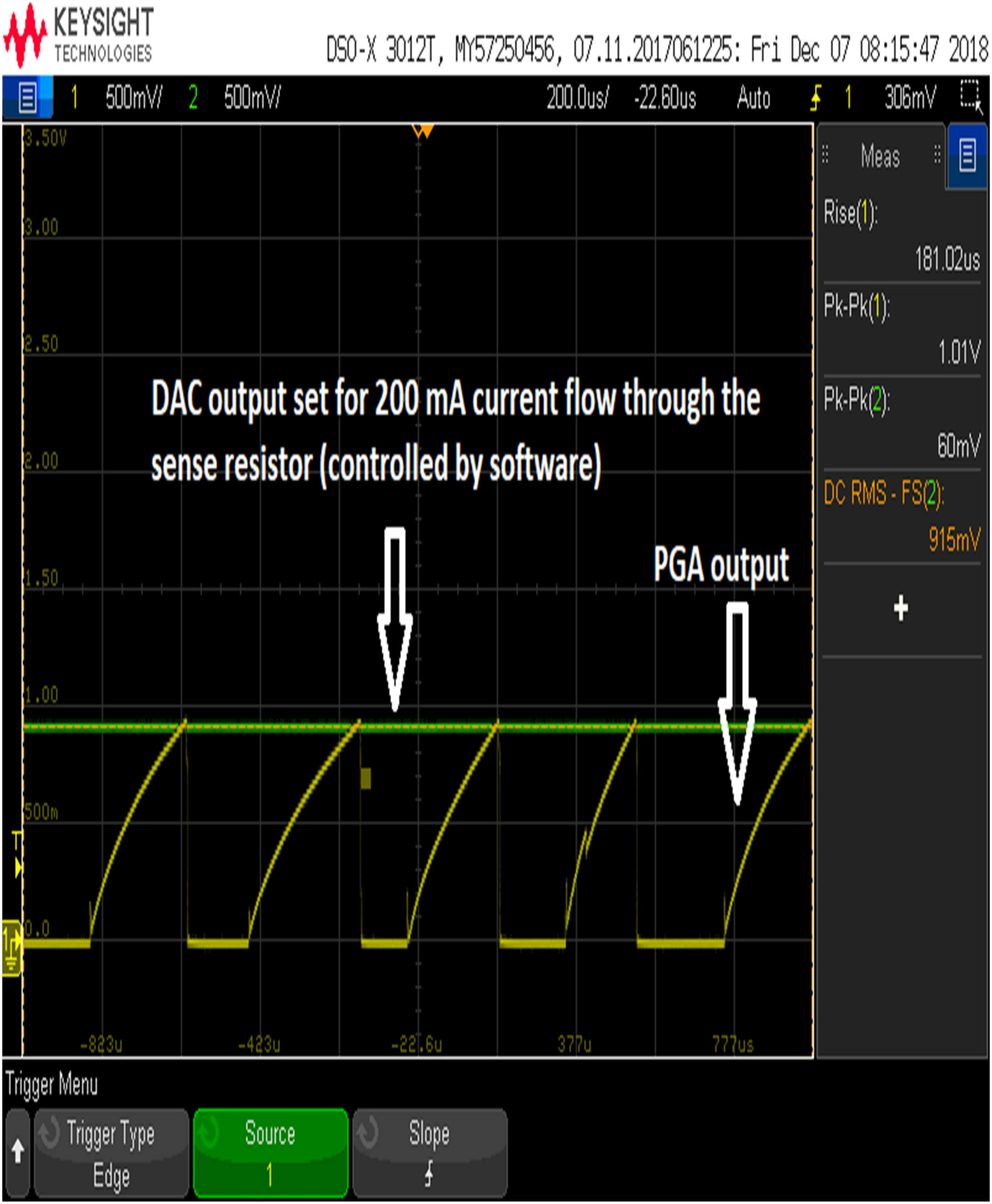
```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/
#include "project.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

int main(void)
{
    CyGlobalIntEnable;
    DAC_Start();
    COMP_Start();
    PGA_Start();
    Clock_1_Start();
    PWM_Start();
    SH_Start();
    ADC_Start();
    LCD_Start();
    LCD_ClearDisplay();

    uint8_t DAC_value;
    uint16_t DAC_mA;
    uint16_t ADC_output, ADC_counts, ADC_counts_mA;

    while(1)
    {
            LCD_Position(0,0);
            LCD_PrintString("DAC: ");
            {
                DAC_value = 30; // for 100 mA
                DAC_SetValue(DAC_value);
                DAC_mA = 100;
                LCD_Position(0,5);
                LCD_PrintNumber(DAC_mA);
                LCD_Position(0,10);
                LCD_PrintString("mA");
```

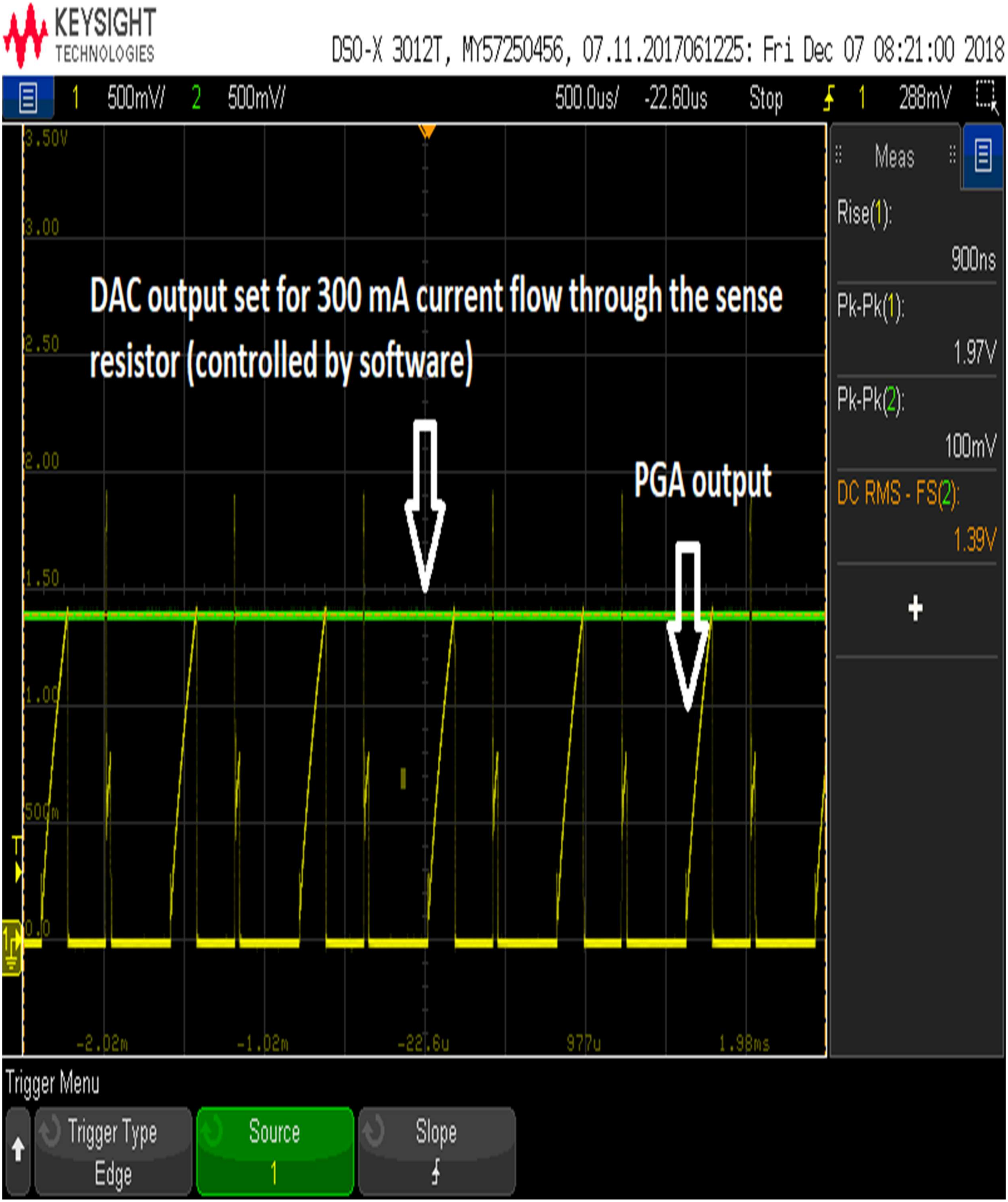
```
                CyDelay(2000);
                {
                    ADC_output = ADC_GetResult16();
                    ADC_counts = ADC_CountsTo_mVolts(ADC_output);
                    ADC_counts_mA = ((ADC_counts)/5);
                    LCD_Position(1,0);
                    LCD_PrintString("ADC: ");
                    LCD_Position(1,5);
                    LCD_PrintNumber(ADC_counts_mA);
                    LCD_Position(1,10);
                    LCD_PrintString("mA");
                }
            }
            {
                DAC_value = 60; // for 200 mA
                DAC_SetValue(DAC_value);
                DAC_mA = 200;
                LCD_Position(0,5);
                LCD_PrintNumber(DAC_mA);
                LCD_Position(0,10);
                LCD_PrintString("mA");
                CyDelay(2000);
                {
                    ADC_output = ADC_GetResult16();
                    ADC_counts = ADC_CountsTo_mVolts(ADC_output);
                    ADC_counts_mA = ((ADC_counts)/5);
                    LCD_Position(1,0);
                    LCD_PrintString("ADC: ");
                    LCD_Position(1,5);
                    LCD_PrintNumber(ADC_counts_mA);
                    LCD_Position(1,10);
                    LCD_PrintString("mA");
                }
            }
            {
                DAC_value = 90; // for 300 mA
                DAC_SetValue(DAC_value);
                DAC_mA = 300;
                LCD_Position(0,5);
                LCD_PrintNumber(DAC_mA);
                LCD_Position(0,10);
                LCD_PrintString("mA");
                CyDelay(2000);
                {
                    ADC_output = ADC_GetResult16();
                    ADC_counts = ADC_CountsTo_mVolts(ADC_output);
                    ADC_counts_mA = ((ADC_counts)/5);
                    LCD_Position(1,0);
                    LCD_PrintString("ADC: ");
                    LCD_Position(1,5);
                    LCD_PrintNumber(ADC_counts_mA);
                    LCD_Position(1,10);
                    LCD_PrintString("mA");
                }
            }
    }
}
```

```
/* [] END OF FILE */
```

E) Place screenshots from your oscilloscope or nScope showing critical loop times or signal outputs. For this lab, you should show plots of the motor current and voltage on one channel, and the FET gate on the other, similar to what is shown in the sample below.


KEYSIGHT TECHNOLOGIES
DSO-X 3012T, MY57250456, 07.11.2017061225: Fri Dec 07 08:16:20 2018
1 500mV/ 2 500mV/
200.0us/ -22.60us Auto 1 306mV
Meas
Rise(1): 31.71us
Pk-Pk(1): 540mV
Pk-Pk(2): 40mV
DC RMS - FS(2): 435mV
DAC output set for 100 mA current flow through the sense resistor (controlled by software)
FET Gate is on
FET Gate is off
PGA output with a gain of 8
3.50V
3.00
2.50
2.00
1.50
1.00
500m
0.0
-823u
-423u
-22.6u
377u
777us
Trigger Menu
Trigger Type Edge
Source 1
Slope


KEYSIGHT TECHNOLOGIES
DSO-X 3012T, MY57250456, 07.11.2017061225: Fri Dec 07 08:15:47 2018
1 500mV/ 2 500mV/ 200.0us/ -22.60us Auto 1 306mV
DAC output set for 200 mA current flow through the sense resistor (controlled by software)
PGA output
Meas
Rise(1): 181.02us
Pk-Pk(1): 1.01V
Pk-Pk(2): 60mV
DC RMS - FS(2): 915mV
3.50V
3.00
2.50
2.00
1.50
1.00
500m
0.0
-823u
-423u
-22.6u
377u
777us
Trigger Menu
Trigger Type
Edge
Source
1
Slope


KEYSIGHT TECHNOLOGIES
DSO-X 3012T, MY57250456, 07.11.2017061225: Fri Dec 07 08:21:00 2018
1 500mV/ 2 500mV/ 500.0us/ -22.60us Stop 1 288mV
DAC output set for 300 mA current flow through the sense resistor (controlled by software)
PGA output
Meas
Rise(1): 900ns
Pk-Pk(1): 1.97V
Pk-Pk(2): 100mV
DC RMS - FS(2): 1.39V
-2.02m -1.02m -22.6u 977u 1.98ms
Trigger Menu
Trigger Type Edge
Source 1
Slope