Matthew Russell, Joshua Ramayrat

Professor Wolfe

ELEN 123, Mechatronics

**Laboratory #7: Positioning Motors**

**Experiment 1**

The following is the arduino code used to control the servo motor with a knob:

*#include <Servo.h>*

*Servo myservo; // create servo object to control a servo*

*int potpin = 0; // analog pin used to connect the potentiometer*

*int val; // variable to read the value from the analog pin*

*void setup() {*

*myservo.attach(9); // attaches the servo on pin 9 to the servo object*

*}*

*void loop() {*

*val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)*

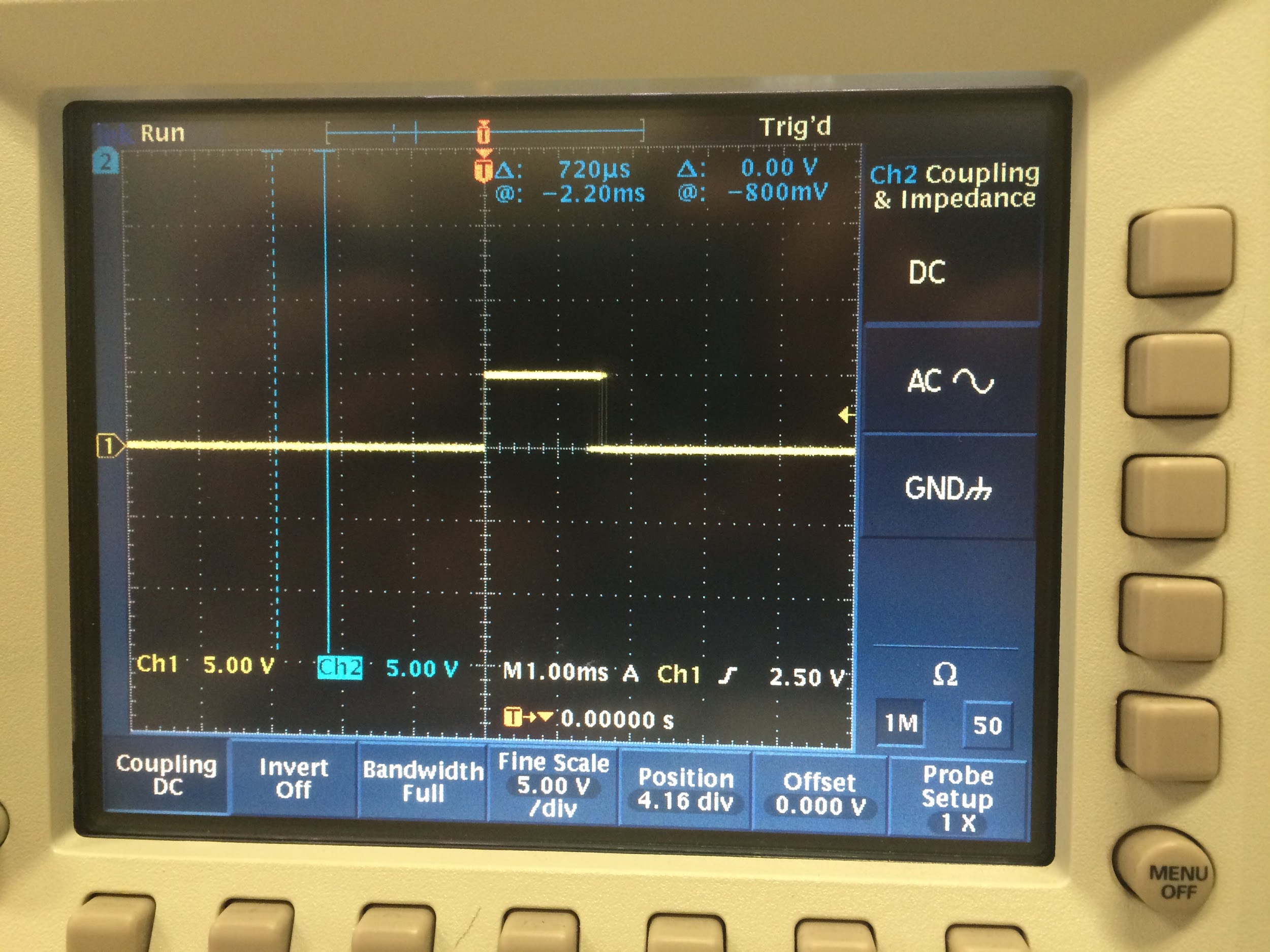
*val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)*

*myservo.write(val); // sets the servo position according to the scaled value*

*delay(15); // waits for the servo to get there*

*}*

5. VA0 = 2.51 V



**Figure 1:** Oscilloscope reading for servo

It was asked to find the voltage reading of the servo when the knob was turned to the very right:

VEoT,Right = 4.51 V

VEoT,Left = .08 V

It was also asked to remap the potentiometer based on the maximum and minimum rotation of the servo arm. The voltage readings yielded 0.8 volts 4.51, implying remapping values of 16 and 923 respectively:

*#include <Servo.h>*

*Servo myservo; // create servo object to control a servo*

*int potpin = 0; // analog pin used to connect the potentiometer*

*int val; // variable to read the value from the analog pin*

*void setup() {*

*myservo.attach(9); // attaches the servo on pin 9 to the servo object*

*}*

*void loop() {*

*val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)*

*val = map(val, 17, 920, 0, 180); // scale it to use it with the servo (value between 0 and 180)*

*myservo.write(val); // sets the servo position according to the scaled value*

*delay(15); // waits for the servo to get there*

}

It was asked to integrate a servo with the accelerometer such that when the car is tilted 30 and -30 degrees, the servo would still be pointing up straight. The code for this is as follows:

*#include <Servo.h>*

*const int xInput = A0;*

*const int yInput = A1;*

*const int zInput = A2;*

*const int buttonPin = 13;*

*// Raw Ranges:*

*// initialize to mid-range and allow calibration to*

*// find the minimum and maximum for each axis*

*int xRawMin = 512;*

*int xRawMax = 512;*

*int yRawMin = 512;*

*int yRawMax = 512;*

*int zRawMin = 512;*

*int zRawMax = 512;*

*// Take multiple samples to reduce noise*

*const int sampleSize = 10;*

*Servo myservo; // create servo object to control a servo*

*int potpin = 0; // analog pin used to connect the potentiometer*

*int val;*

*void setup()*

*{*

*analogReference(EXTERNAL);*

*Serial.begin(9600);*

*analogReference(EXTERNAL);*

*myservo.attach(22);*

*}*

*void loop()*

*{*

*int xRaw = ReadAxis(xInput);*

*int yRaw = ReadAxis(yInput);*

*int zRaw = ReadAxis(zInput);*

*if (digitalRead(buttonPin) == LOW)*

*{*

*AutoCalibrate(xRaw, yRaw, zRaw);*

*}*

*else*

*{*

*Serial.print("Raw Ranges: X: ");*

*Serial.print(xRawMin);*

*Serial.print("-");*

*Serial.print(xRawMax);*

*Serial.print(", Y: ");*

*Serial.print(yRawMin);*

*Serial.print("-");*

*Serial.print(yRawMax);*

*Serial.print(", Z: ");*

*Serial.print(zRawMin);*

*Serial.print("-");*

*Serial.print(zRawMax);*

*Serial.println();*

*Serial.print(xRaw);*

*Serial.print(", ");*

*Serial.print(yRaw);*

*Serial.print(", ");*

*Serial.print(zRaw);*

*// Convert raw values to 'milli-Gs"*

*long xScaled = map(xRaw, xRawMin, xRawMax, -1000, 1000);*

*long yScaled = map(yRaw, yRawMin, yRawMax, -1000, 1000);*

*long zScaled = map(zRaw, zRawMin, zRawMax, -1000, 1000);*

*// re-scale to fractional Gs*

*float xAccel = xScaled / 1000.0;*

*float yAccel = yScaled / 1000.0;*

*float zAccel = zScaled / 1000.0;*

*Serial.print(" :: ");*

*Serial.print(xAccel);*

*Serial.print("G, ");*

*Serial.print(yAccel);*

*Serial.print("G, ");*

*Serial.print(zAccel);*

*Serial.println("G");*

*delay(500);*

*}*

*val = analogRead(A1); // reads the value of the potentiometer (value between 0 and 1023)*

*val = map(val, 470, 540, 45, 105); // scale it to use it with the servo (value between 0 and 180)*

*myservo.write(val); // sets the servo position according to the scaled value*

*delay(15);*

*}*

*//*

*// Read "sampleSize" samples and report the average*

*//*

*int ReadAxis(int axisPin)*

*{*

*long reading = 0;*

*analogRead(axisPin);*

*delay(1);*

*for (int i = 0; i < sampleSize; i++)*

*{*

*reading += analogRead(axisPin);*

*}*

*return reading/sampleSize;*

*}*

*//*

*// Find the extreme raw readings from each axis*

*//*

*void AutoCalibrate(int xRaw, int yRaw, int zRaw)*

*{*

*Serial.println("Calibrate");*

*if (xRaw < xRawMin)*

*{*

*xRawMin = xRaw;*

*}*

*if (xRaw > xRawMax)*

*{*

*xRawMax = xRaw;*

*}*

*if (yRaw < yRawMin)*

*{*

*yRawMin = yRaw;*

*}*

*if (yRaw > yRawMax)*

*{*

*yRawMax = yRaw;*

*}*

*if (zRaw < zRawMin)*

*{*

*zRawMin = zRaw;*

*}*

*if (zRaw > zRawMax)*

*{*

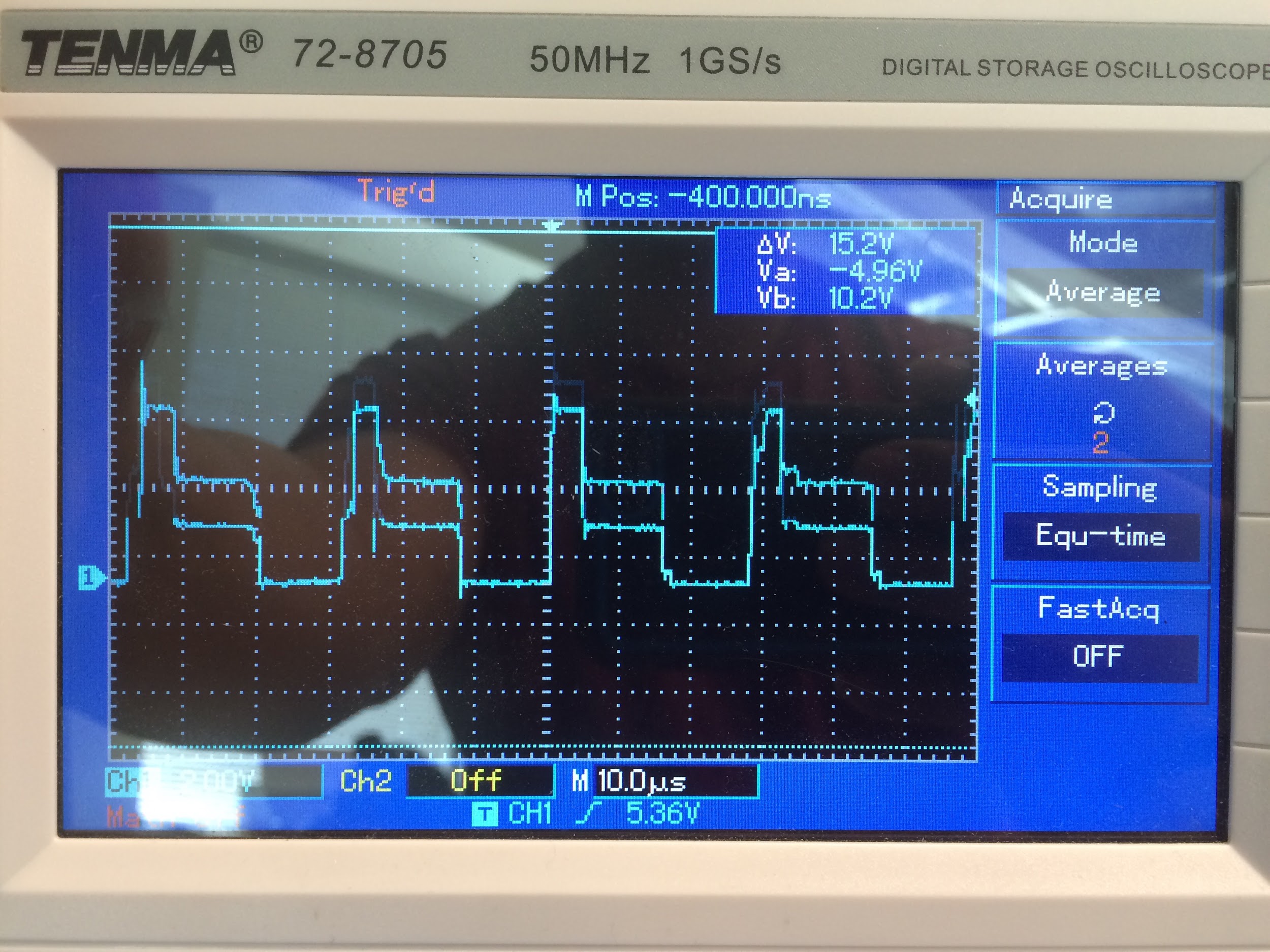
*zRawMax = zRaw;*

*}*

*}*

**Task 2:**

1. It was asked to record the waveform between motor terminal A+ and A- to ground using DC coupling:



**Figure 1:** Waveform of 6V DC power to the stepper motor.

1. From the preceding figure, the most power consumed is from 0 to 5 micro seconds.
2. The following program was used to microstep the motor 384 steps forward and 384 steps backwards:

*int DIR = 8;*

*int STEP = 9;*

*void setup(){*

*pinMode(DIR,OUTPUT);*

*pinMode(STEP, OUTPUT);*

*digitalWrite(DIR, LOW);*

*digitalWrite(STEP, LOW);*

*for (int i = 0; i < 384; i++){*

*digitalWrite(STEP, HIGH);*

*delay(1);*

*digitalWrite(STEP, LOW);*

*delay(1);*

*}*

*digitalWrite(DIR, HIGH);*

*for (int i = 0; i < 384; i++){*

*digitalWrite(STEP, HIGH);*

*delay(1);*

*digitalWrite(STEP, LOW);*

*delay(1);*

*}*

*}*

*void loop() {}*