TERRAN BLAKE // ECE 241 // Lab 10 // Wednesday 7:30am

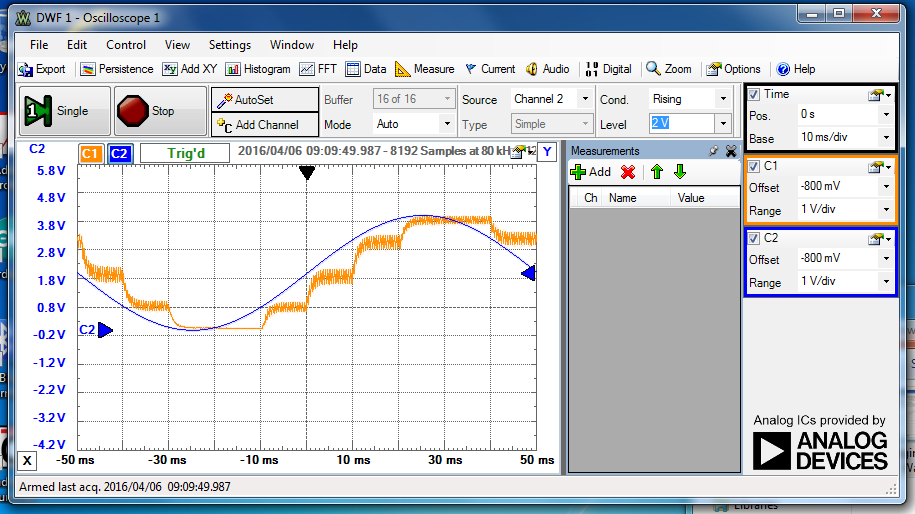


Image 1: Original Sine Wave at 10hz

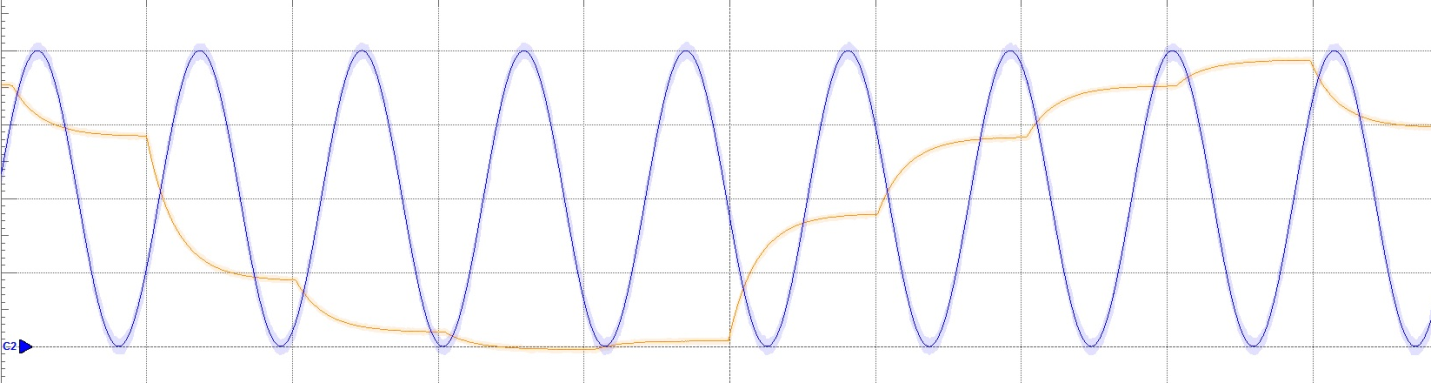


Image 2: Original Sine Wave at 90hz

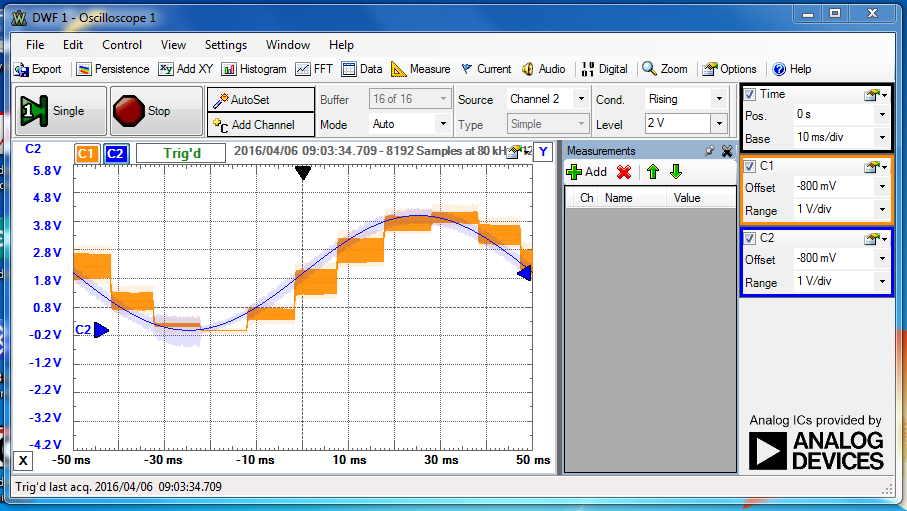


Image 3: Sine Wave with 0.2uf capacitor and 1k resistor

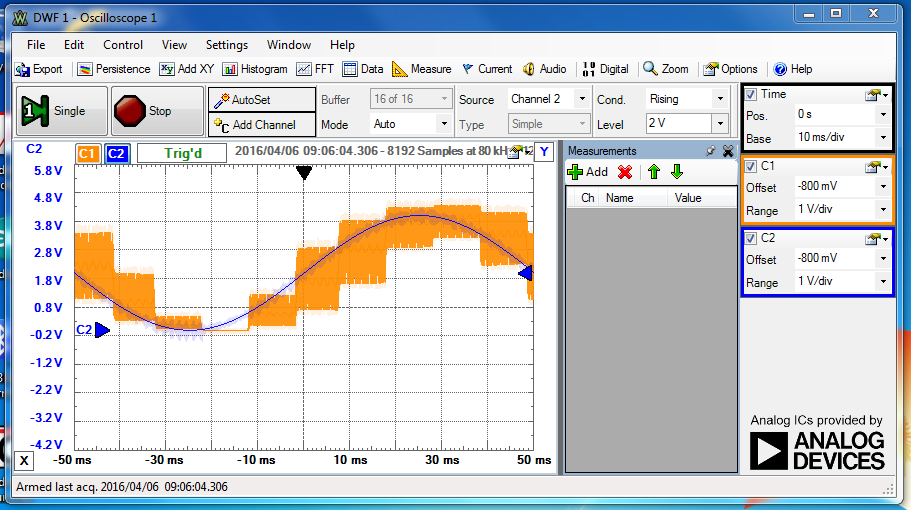


Image 4: Sine Wave with PWM of 250 microseconds 10hz

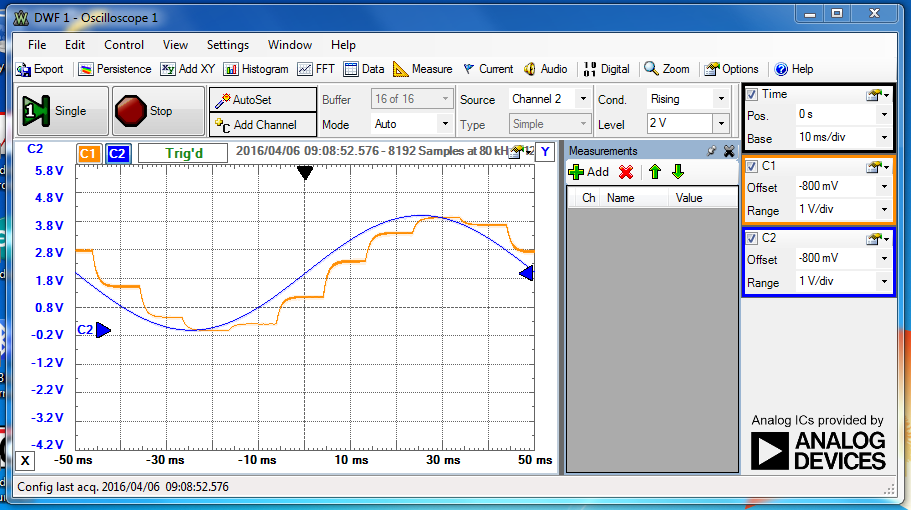


Image 5: Sine Wave with 10k resistor, 0.1uf capacitor, PWM at 100 microseconds, and a 10hz frequency

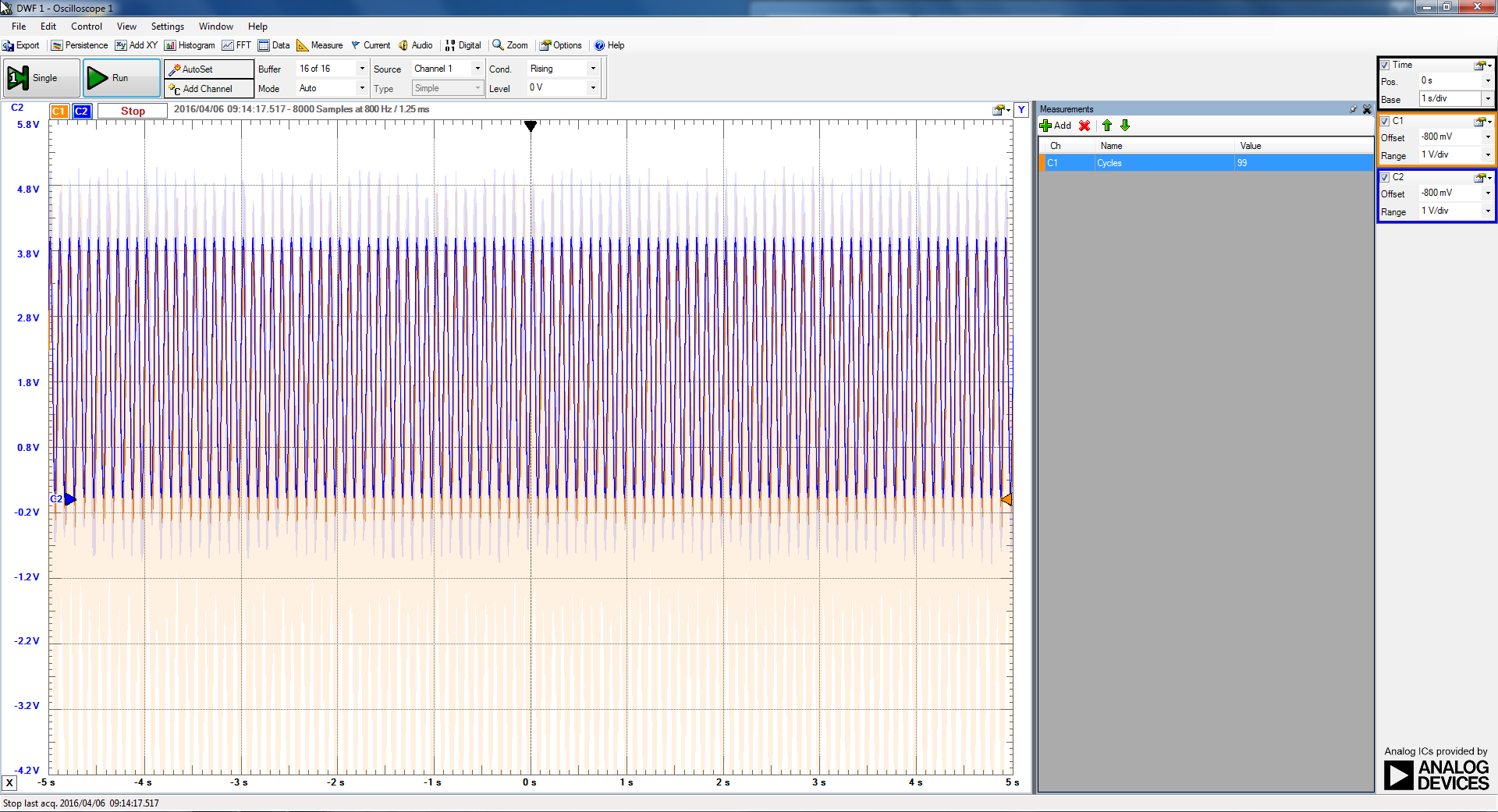


Image 6: Frequency of signal for Image 5

Questions:

1. The period had the most impact on the “cleanness” of the output waveform, because the Arduino can only process information at a certain rate, which shows when the frequency of the wave is turned up.
2. I achieved a 99 / 10Hz sampling rate, as seen in image 6. So it was very close to what it should have been.

#include <TimerOne.h>;

int ledPin = 13; //pin for the led

unsigned long Timer2 = 0;

const long Timer2Interval = 10; //values for the timer

unsigned long currentMillis = millis();

const long DutyCycle = 551; //value for the PWM, has been edited for parts of lab

void setup() {

pinMode (13,OUTPUT); //pin modes for the reading

Timer1.initialize(100);

Timer1.pwm(9, 0, 100);

// Serial.begin(9600); //was used as a test to check timer

// Serial.available();

}

void loop() {

if( millis() - Timer2 > Timer2Interval) { //keeps the program in a 10ms interval

Timer2 = Timer2 + Timer2Interval;

bitSet(PORTB,5); //saves the frequency

Timer1.setPwmDuty(9, analogRead(A0)); //reads the frequency

bitClear(PORTB,5);

// Serial.print("hello"); // test for the timer

}

}