RESISTANCE TO FREQUENCY CONVERTOR

A Project Report

Submitted by

ARAVINDAKSHAN S

In partial fulfilment of the requirements

For the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **RESISTANCE TO FREQUENCY CONVERTOR**, submitted by **ARAVINDAKSHAN S**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr.Boby George
Project Guide
Associate Professor
Dept. of Electrical Engg.
IIT-Madras, 600 036

Place: Chennai
Date: MAY 2017

ACKNOWLEDGEMENTS

I am thankful to **Dr. Boby George** for giving me the opportunity to work with him and showing confidence in me. I would like to gratefully acknowledge him for providing direction to this work. I have been immensely benefited by his constant guidance, support and motivation. I would also like to gratefully acknowledge **Dr.Jagadeesh Kumar** for his valuable insights during the review meetings.

I am deeply indebted to my mentor **Sreenath** for the constant help, discussions and advice, by which I benefitted a lot. I would also thank my lab mates **Deb Jyoti**, **Ram** and **Dinesh** for all the useful inputs on how to use the various equipments in the lab by which I was able to start my project.

At the end, I would like to thank my parents who have always been kind and working hard to my growth. Whatever I am today is due to their hard work and blessings.

ABSTRACT

The proposed technique presented in this thesis uses an integral controller and a voltage to frequency convertor in negative feedback. A switched capacitive circuit that emulates a resistor is employed in this scheme. The value of the emulated resistor changes as a function of the frequency at which the switch is operated. The circuit stabilizes in negative feedback only if the equivalent resistance of the switched capacitive circuit at the current frequency is same as that of the unknown resistance. This gives us a relation between the stabilization frequency and the unknown resistance, from which the unknown resistance can be calculated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

a) Problems in conventional measurement:

Resistive sensors are used in a variety of situations. Quick and accurate measurement may be vital in several situations. For example, they are used in resistive load cells. Resistive load cells work on the principle of piezo-resistivity. When a load/force/stress is applied to the sensor, it changes its resistance. This change in resistance leads to a change in output voltage when an input voltage is applied [1]. We need accurate measurements to measure such small changes in resistance.

Standard resistance measurement using a multimeter suffers from several drawbacks. The ammeter coil might have non zero resistance. This might lead to a voltage drop across it, thus changing the desired readings. Or else, the voltmeter coil might have a finite resistance. This might lead to a part of the current flowing through the voltmeter thus again changing the readings. The relative errors in both these readings add up leading to even more error in the resistance value.

b) Proposed method:

The method used here uses the technique of negative feedback.

When the circuit is turned on, the frequency of the VCO (Voltage Controlled Oscillator) will correspond to the initial value of the voltage (V_1). During the ON periods of the square wave, the voltage across the capacitor becomes equal to V_r. The same charge flows through C_f. Resistance R_x conducts a constant current of V_r/R_x. Thus the circuit stabilizes in negative feedback if the total charge flown through C_s in one time period is same as the charge through R_x in one time period. This condition would mean the effective resistance of the capacitance-switch combination is same as that of the unknown resistance.

c) <u>Implementation</u>:

The switch used was MAX4053 which has 3 SPDT (Singe pole double throw) switches. For voltage to frequency conversion, LM331 was used. A particular set of connection allows it to be used as a VCO (shown in diagram later). The op-amps used were OP07CP. The voltage regulator used was LM385Z-1.2. It provides a near constant voltage of 1.2V. Arduino Uno was used to count the rising edges of the resulting square wave to determine the frequency.

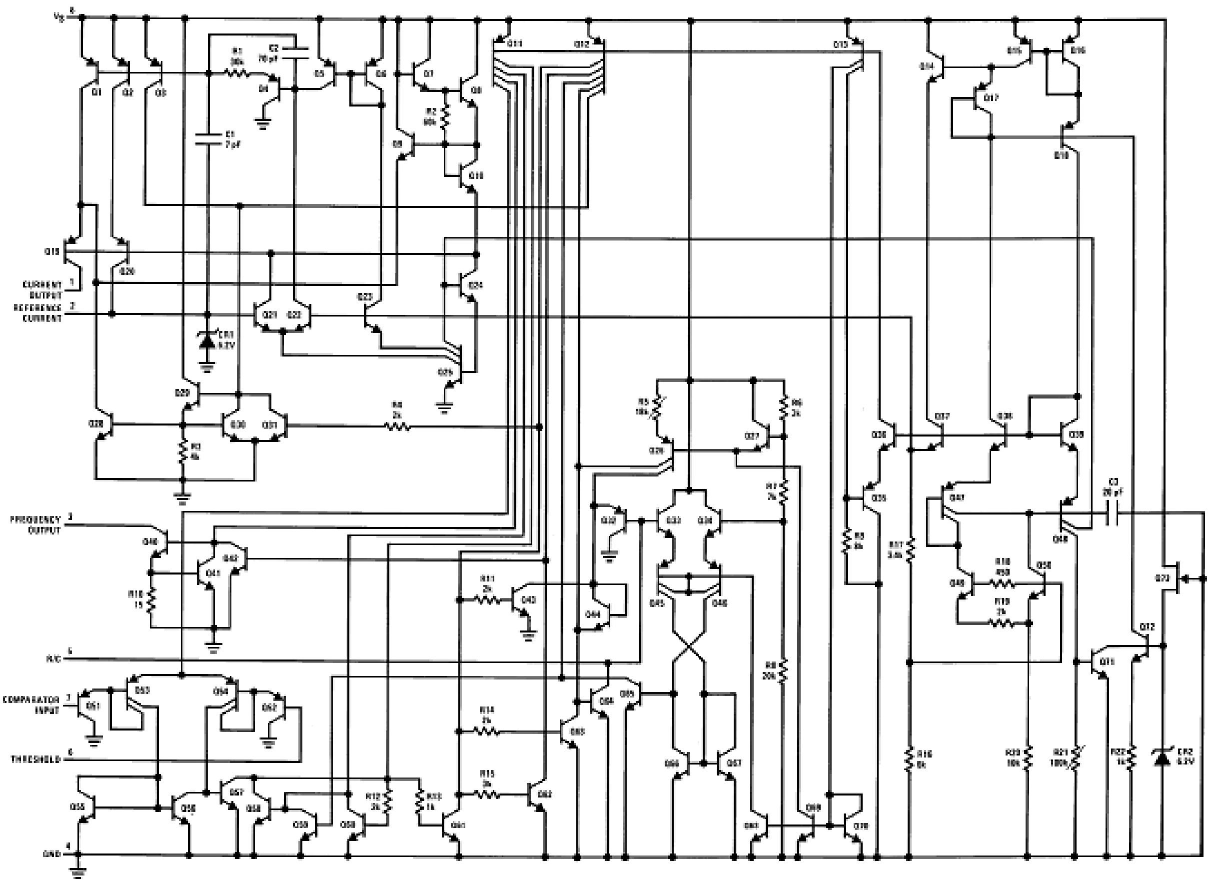

Figure 1: Schematic diagram of LM331

2 RESISTANCE TO FREQUENCY CONVERTOR

This is diagram of the circuit used.

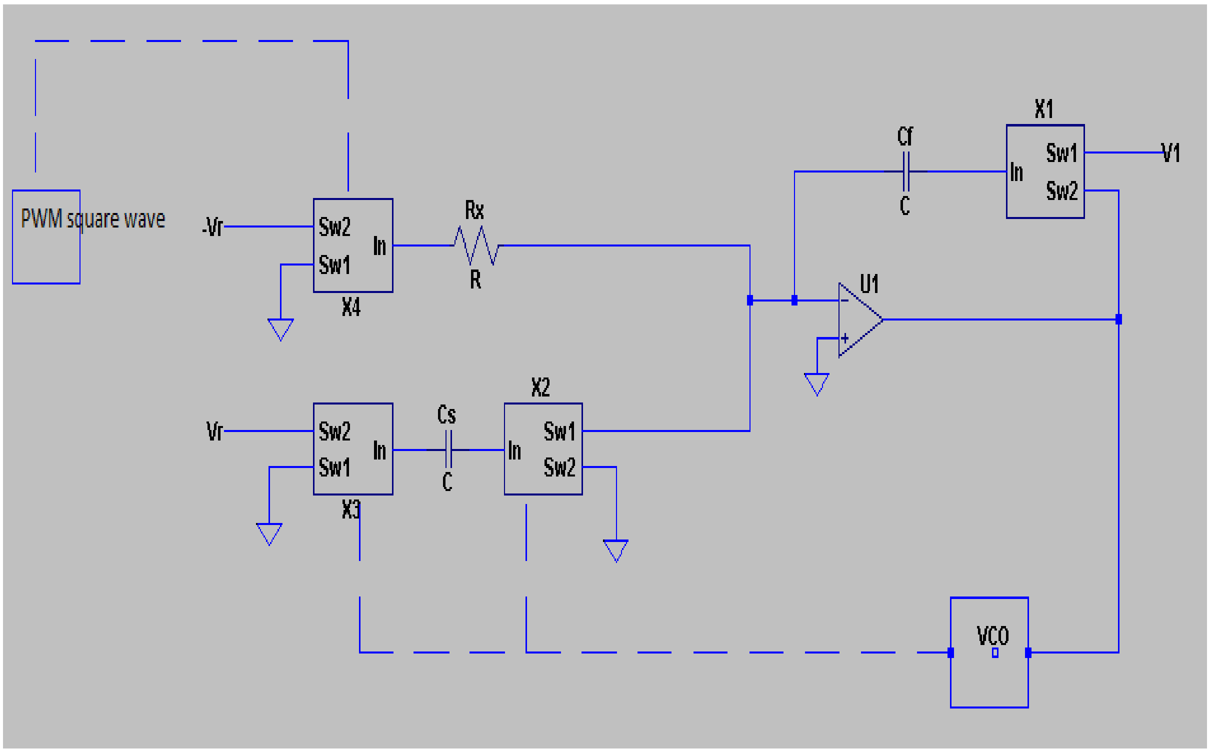

Figure 2: The circuit

In the above diagram, the circuit functions in negative feedback. Feedback stabilization is achieved when the charge flowing through C_S in one time period is same as that flowing through R_X in one time period. The frequency of the VCO output at which this happens has one-one relation with the unknown resistance and thus it allows for the calculation of the unknown resistance. The effective resistance of the capacitance would be equal to that of the unknown resistance.

a)<u>Working of the circuit</u>:

The circuit achieves feedback stabilization when the charge passing through the capacitor C_S in one time period of the signal (the output from the VCO) is same as the charge passing through the resistor R_X in one period. At the rising edge of the signal, a charge equal to $V_R C_S$ passes through the capacitor C_S. The same charge passes through C_F since the intermediate point is at virtual ground due to negative feedback. A constant current equal to V_R/R_X is passing through the resistor R_X all the time during both the HIGH and LOW states. The total current flown through the resistor in one time period, assuming frequency at feedback stabilization is f, is $V_R/(R_X f)$. This should be equal to $V_R C_S$. Thus the unknown resistance is

$$R_X=1/(C_S f)$$

b) Component description:

(i) *Switch (MAX4053):*

It has three SPDT (Single Pole Double Throw) switches. V^+ can be anything between 2.7V and 8V; V^- anywhere between -2.7V and -8V.The diagram as taken from the datasheet is shown below.

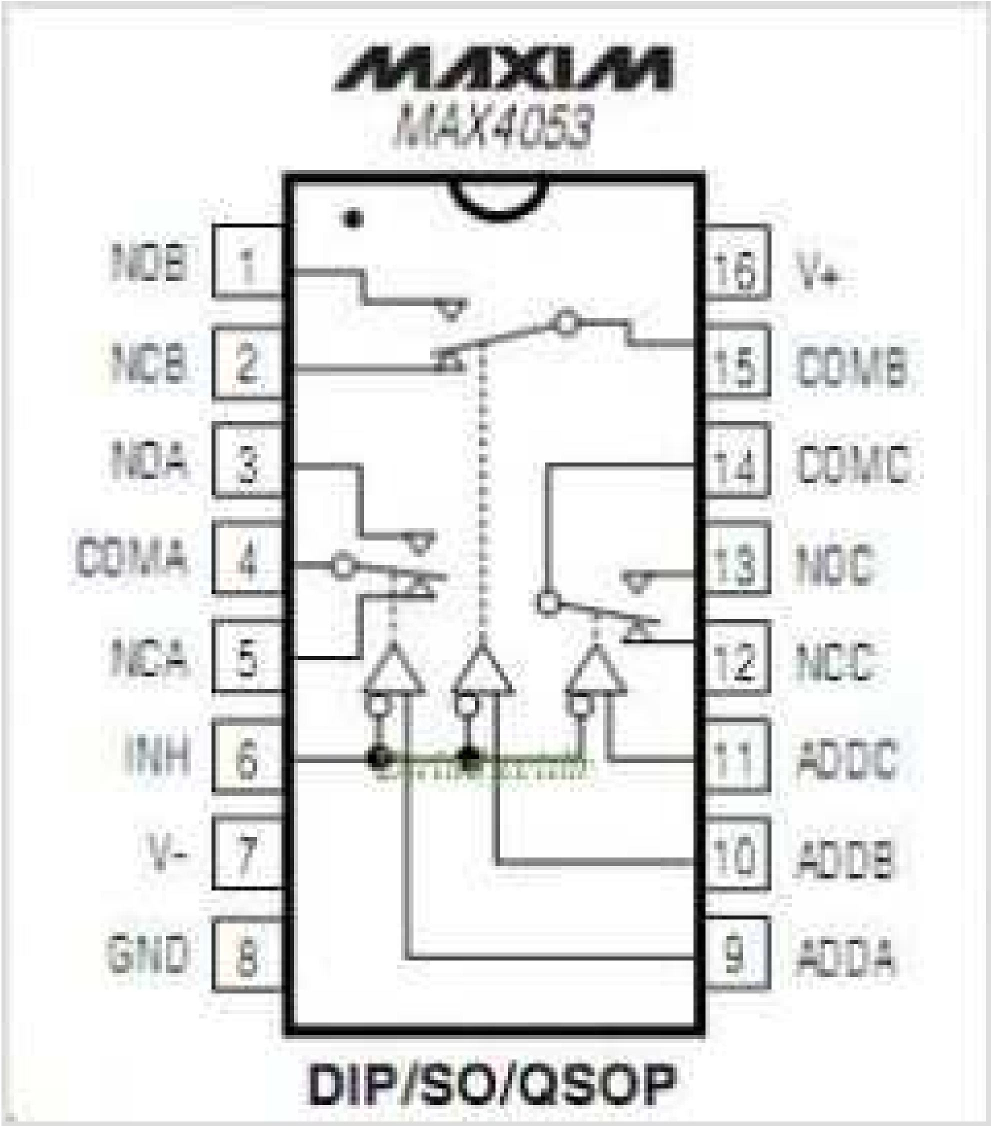

Figure 3: MAX4053 pin diagram [2]

INH	ADDC	ADDB	ADDA	Connections
1	X	X	x	All switches open
0	0	0	0	COMA-NCA COMB-NCB COMC-NCC
0	0	0	1	COMA-NOA COMB-NCB COMC-NCC
0	0	1	0	COMA-NCA COMB-NOB COMC-NCC
0	0	1	1	COMA-NOA COMB-NOB COMC-NCC
0	1	0	0	COMA-NCA COMB-NCB COMC-NOC
0	1	0	1	COMA-NOA COMB-NCB COMC-NOC
0	1	1	0	COMA-NCA COMB-NOB COMC-NOC
0	1	1	1	COMA-NOA COMB-NOB COMC-NOC

Table 1: Logical connection in MAX4053 [3]

(ii) VFC -Voltage to Frequency Convertor (LM331):

The following details and diagrams are taken from the datasheet of the IC.[4]

The LMx31 family of voltage-to-frequency converters are ideally suited for use in simple low-cost circuits for analog-to-digital conversion, precision frequency- to-voltage conversion, long-term integration, linear frequency modulation or demodulation, and many other functions. The output when used as a voltage-to-frequency converter is a pulse train at a frequency precisely proportional to the applied input voltage. Thus, it provides all the inherent advantages of the voltage-to-frequency conversion techniques, and is easy to apply in all standard voltage-to-frequency converter applications.

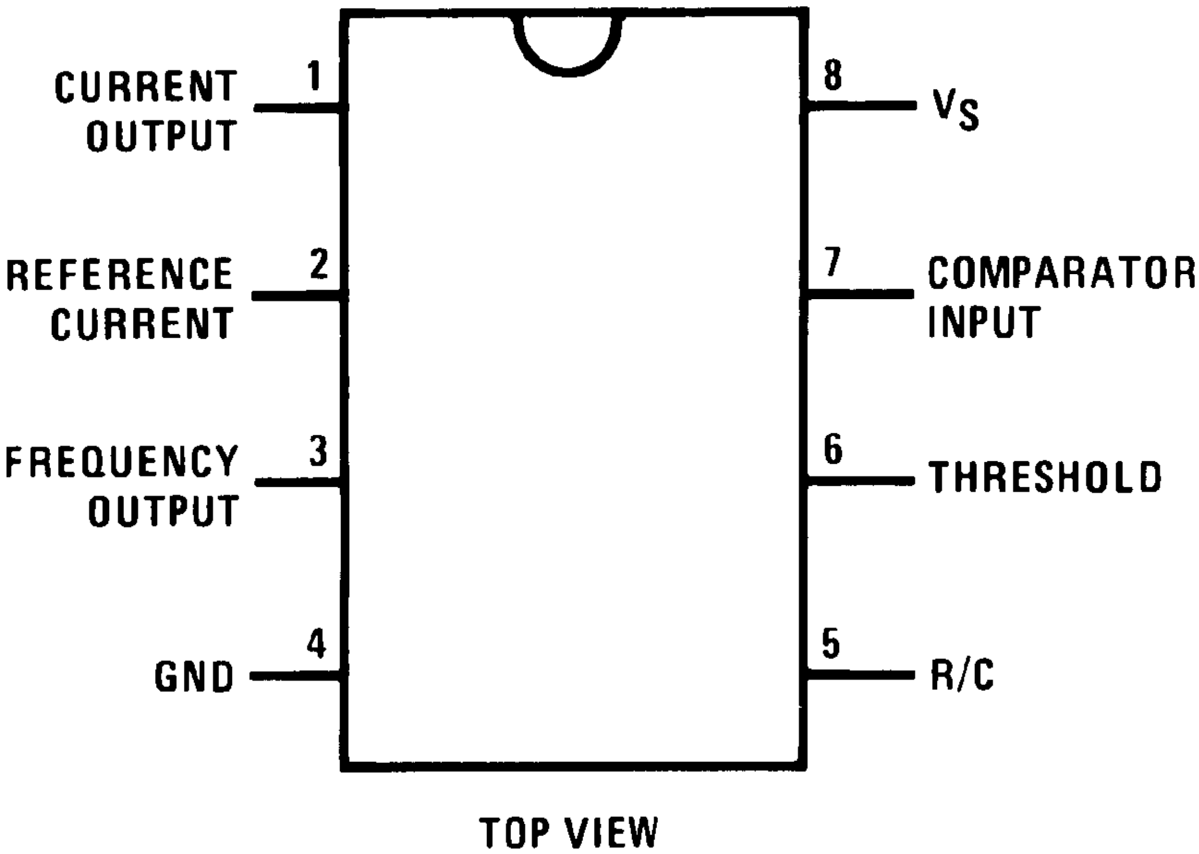

Figure 4: LM331 pin diagram

The simple stand-alone V-to-F converter is shown in the figure below

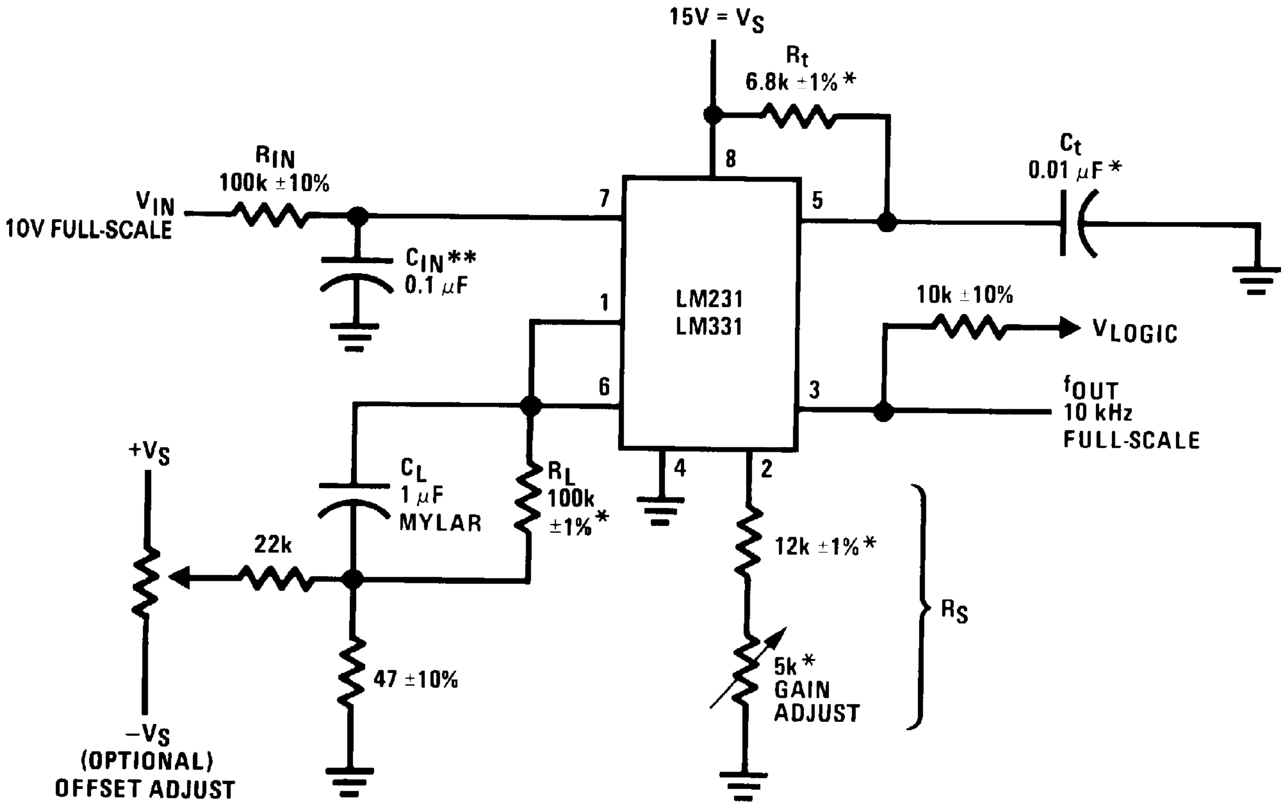

Figure 5: LM331-Connections to use it as VFC

Design Procedure:

A capacitor CIN is added from pin 7 to ground to act as a filter for VIN, use of a 0.1 µF is appropriate for this application. A value of 0.01 µF to 0.1 µF will be adequate in most cases; however, in cases where better filtering is required, a 1µF capacitor can be used. When the RC time constants are matched at pin 6 and pin 7, a voltage step at VIN will cause a step change in f_{OUT}. If C_{IN} is much less than C_L, a step at V_{IN} may cause f_{OUT} to stop momentarily. Next, we cancel the comparator bias current by setting R_{IN} to 100 kΩ to match R_L. This will help to minimize any frequency offset. For best results, all the components should be stable low-temperature-coefficient components, such as metal-film resistors. The capacitor should have low dielectric absorption; depending on the temperature characteristics desired, NPO ceramic, polystyrene, Teflon or polypropylene is best suited. The resistance RS at pin 2 is made up of a 12-kΩ fixed resistor plus a 5-kΩ (cermet, preferably) gain adjust rheostat. The function of this adjustment is to trim out the gain tolerance of the LMx31, and the tolerance of R_T, R_L and C_T. A 47-Ω resistor in series with the 1-µF capacitor (CL) provides hysteresis, which helps the input comparator provide the excellent linearity. This results in the transfer function of

$$f_{OUT} = (V_{IN} / 2.09\ V) \times (R_S / R_L) \times (1 / R_T C_T) = 1196 V_{IN} \text{(for given values)}$$

(iii) OP-AMP(OP07CP):

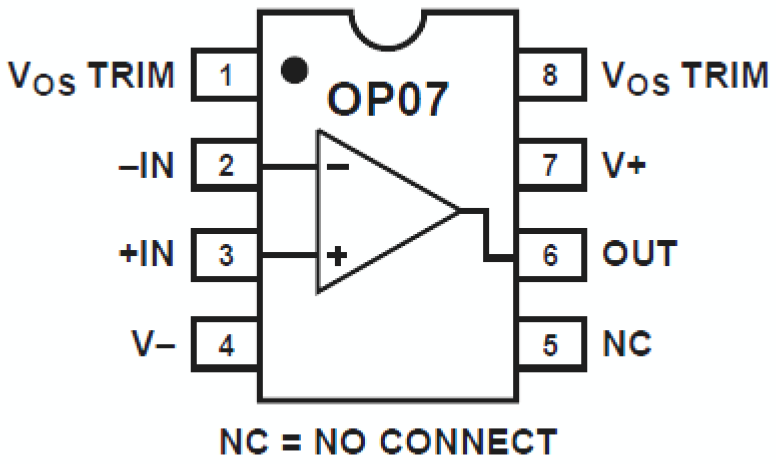

Figure 6: OP07CP pin diagram

c) Measurement of frequency:

To measure frequency accurately, Arduino Uno is used to find the time interval between two rising edges. The occurrence of a rising edge is detected [5], and an interrupt is generated [6] to count it. The time for 1000 rising edges to occur is found. That count divided by 1000 gives the time period of the signal. The reciprocal of time period gives frequency. See Appendix Part A for the code.

The code generates an interrupt every time a rising edge occurs. The time period is determined from this and thus so is the frequency.

d) Improving range of resistance measurement:

A PWM signal can be used on V_R with a variable duty cycle. Since the VCO's input is the output of the op-amp it cannot go beyond the saturation voltage of the op-amp (5V in our case). This results in an upper limit to the output frequency. Thus, this causes a lower limit to the resistance we can measure. This limit can be improved by reducing the duty cycle of the PWM when the frequency touches the upper limit. This reduces the value of the constant current flowing through the resistance and thus results in a higher time period, i.e., lower frequency. To achieve this, the following code was used to generate a PWM signal at a frequency much higher than the frequency of operation [7], but due to clock frequency constraints the code generates only up to 9.5 KHz, which is not enough. See Appendix Part B for the code.

3 OBSERVATIONS AND RESULTS

The circuit connection for the diagram shown previously is shown below.

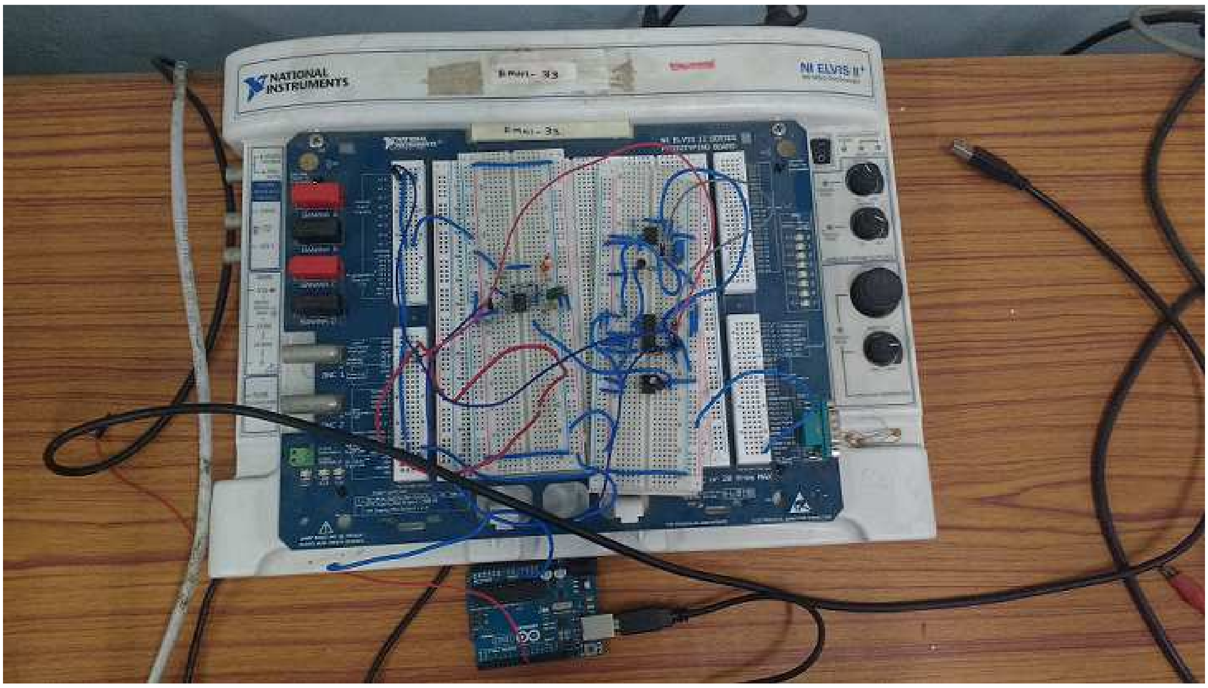

Figure 7: Picture of the circuit

The frequency was measured initially using NI ELVIS and then arduino counters were used later. The plots of resistance versus frequency are expected to show as hyperbolic trend as they are inversely related to each other. This is shown in figure 8. The plots of resistance versus time period are expected to show a linear trend as they are directly related to each other. This is shown in the figure 9.

Here is a set of readings taken.

Resistance(in KΩs)	Frequency(in KHz)
1.38	4.47
2.44	3
3.3	2.18
4.58	1.56
6.01	1.27
8.97	0.875
14.38	0.56

Table 2: Resistance vs Frequency

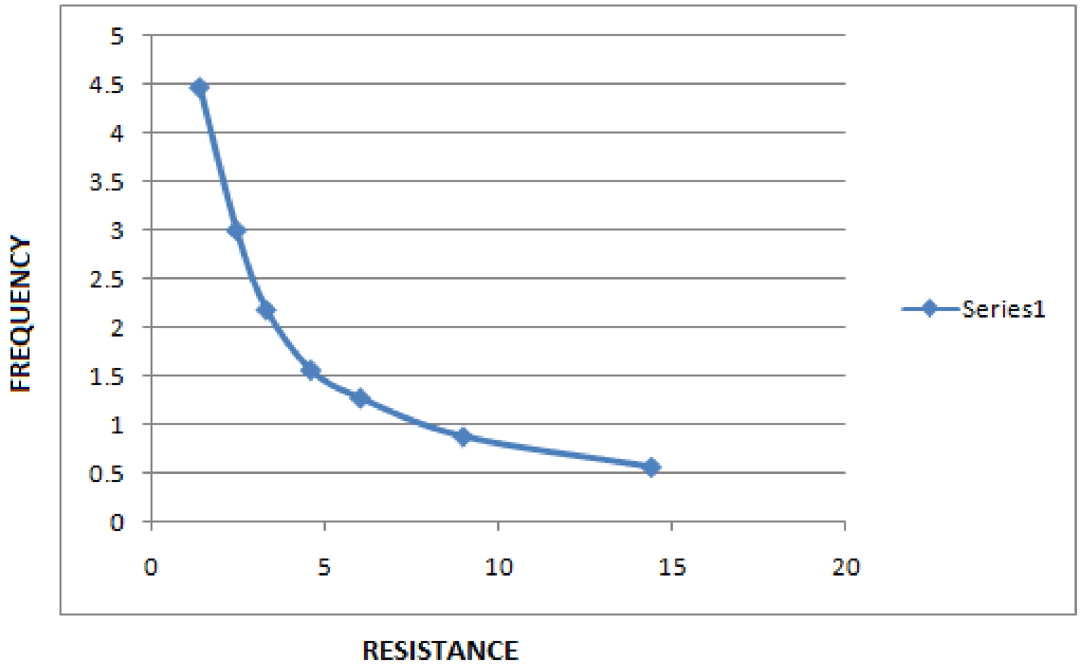

Figure 8: Frequency vs Resistance Graph

Resistance(in KΩs)	Time Period(in µs)
1.38	223.7
2.44	333.3
3.3	458.7
4.58	641.0
6.01	787.4
8.97	1142.9
14.38	1785.7

Table 3: Resistance vs Time period

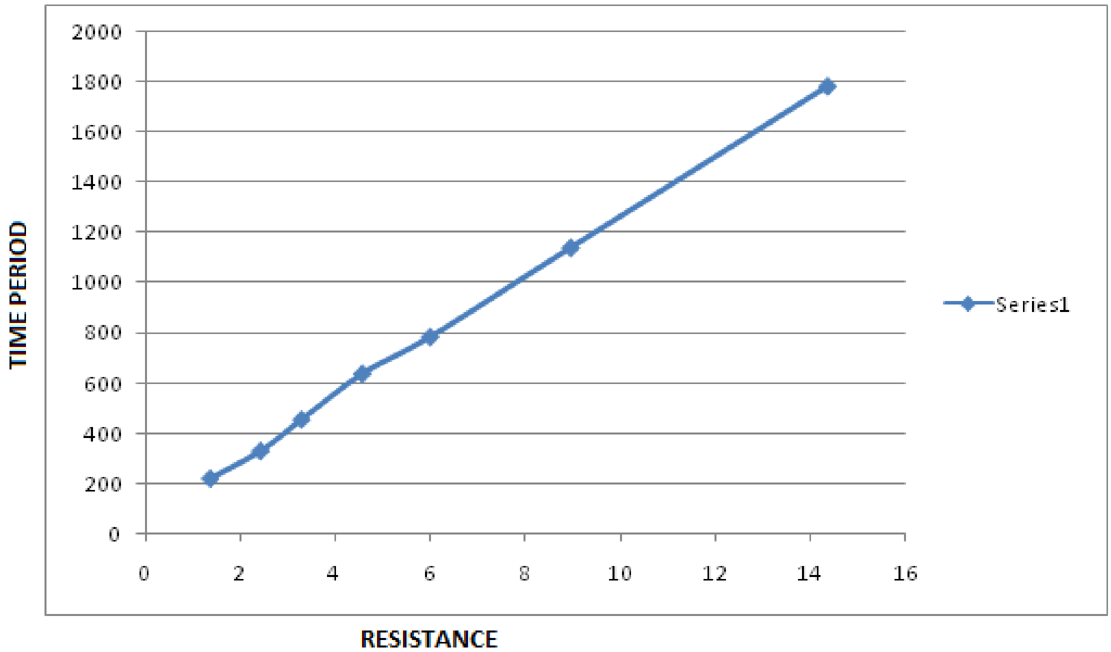

Figure 9: Resistance vs Time period Graph

As can be seen, the expected trend is followed to a high degree of accuracy.

Screenshot of the frequency measurement in NI ELVIS is shown in figure 10 and in arduino is shown in figure 11.

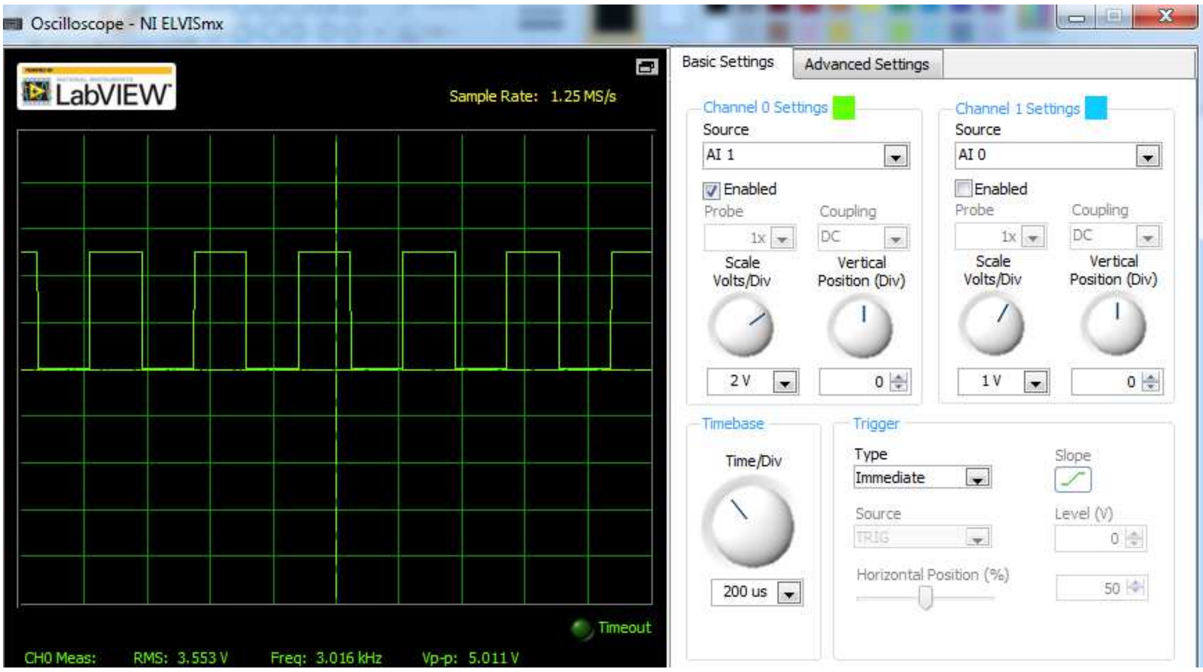

Figure 10: Sample ELVIS output

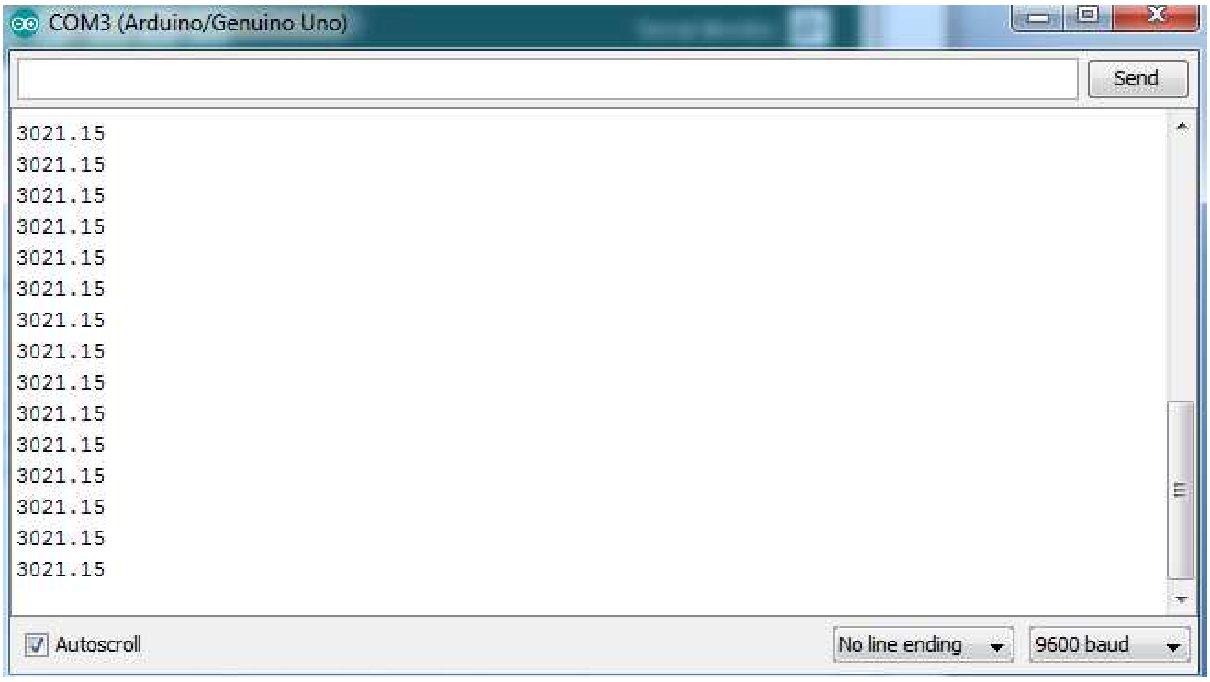

Figure 11: Sample arduino output

4 FUTURE WORKS

Here we have the capability of measuring resistances from 1KΩ-15 KΩ. Getting the PWM controlled input voltage to work would let us extend the range further. So would the usage of a VCF IC with a better range. The slight errors in linearity found here would be removed if the circuit is fabricated as an IC.

REFERENCES

[1] http://www.loadstarsensors.com/what-is-a-load-cell.html

[2] http://circuits.datasheetdir.com/168/MAX4053A-pinout.jpg

[3] https://datasheets.maximintegrated.com/en/ds/MAX4051-MAX4053A.pdf

[4] http://www.ti.com/product/LM331/datasheet

[5] https://www.arduino.cc/en/Tutorial/StateChangeDetection

[6] https://www.arduino.cc/en/Reference/AttachInterrupt

[7] http://www.instructables.com/id/Arduino-Timer-Interrupts/

APPENDIX

Part A: Arduino code to measure frequency

```
volatile unsigned long prev_edge=0;  //variable to store time of previous rising  edge
volatile unsigned long curr_edge=0;  //variable to store current edge
volatile unsigned long timeperiod=0;//variable to store time period
volatile unsigned counter=0;             // to count number of timeperiods crossed
void setup()
{
  pinMode(2, INPUT);                        //Set pin 2 as input
  attachInterrupt(digitalPinToInterrupt(2), measure, RISING);
  //Predefined function to create an interrupt whenever an rising edge occurs.
  Serial.begin(9600);  // begin serial communication
}
void loop()
{
  interrupts();            // allow interrupts
  curr_edge= micros();   //store current time in microseconds
 }
void measure()
 {
   counter++;       //Control is here because the code has encountered an interrupt
   //i.e. a rising edge in this case. So increment counter
   if(counter==1000)  //To measure averaged time period over 1000 cycles to
   //ignore minor fluctuations.
  {
     timeperiod=curr_edge-prev_edge;
    timeperiod/=1000;  //Calculating timeperiod
     Serial.println(1000000/float(timeperiod));  // Printing time period to serial monitor
     counter=0;      //Reset variable for next measurement
```

```
    timeperiod=0; //Reset variable for next measurement
    prev_edge=curr_edge; //Reset variable for next measurement
  }
}
```

Part B: Arduino Code to generate a PWM signal

```
unsigned d=6;         //duty cycle(any integer between 1-10 can be given)
unsigned c=1;         //variable to count the occurrence one-tenth of the time period
void setup()
{
  pinMode(9, OUTPUT);  // setting pin 9 for output
 cli();                          //stop interrupts

  TCCR2A = 0;// set entire TCCR2A register to 0
  TCCR2B = 0;// same for TCCR2B
  TCNT2  = 0;//initialize counter value to 0
  // set compare match register for 400khz increments
  OCR2A = 39;// = (16*10^6) / (40000*1*10) - 1 (must be <256)
  // turn on CTC mode
  TCCR2A |= (1 << WGM21);
  // Set CS20 bit for 1 prescaler
  TCCR2B |= (1 << CS20);
  // enable timer compare interrupt
  TIMSK2 |= (1 << OCIE2A);

sei();//allow interrupts
}//end setup
```

```
ISR(TIMER2_COMPA_vect)
{
  if(c>10)  c=1;
  if(c<=d)    digitalWrite(9,HIGH);
  else digitalWrite(9,LOW);
  c++;
}
void loop()
{
  //do other things here
}
```