AN ABSTRACT OF THE THESIS OF

Estefania Elorriaga Montenegro for the degree of Master of Science in Biological and Ecological Engineering presented on June 15, 2012.

Title: High Frequency Water Vapor Density Measurements using the Beat Frequency Method.

This document describes the design and deployment of a first generation water vapor density sensing unit, the HumiSense. This device is based on an open, air-filled capacitor which is part of a resonant circuit. The frequency of the resonant circuit mixed with a fixed frequency oscillator is the basis of the method to generate a signal that is associated to the change in water vapor density within the open capacitor with time. The physical testing results were inconclusive given that there were many unresolved artifacts in the data. Several suggestions for improving the device for future device generations were provided.

Abstract approved:

_____

Major I. Professor John S. Selker

High Frequency Water Vapor Density Measurements using the Beat Frequency Method

by

Estefania Elorriaga Montenegro

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 15, 2012
Commencement June 2013

Master of Science thesis of Estefania Elorriaga Montenegro presented on June 15, 2012.

APPROVED:

Major Professor, representing Biological and Ecological Engineering

Head of the Department of Biological and Ecological Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Estefania Elorriaga Montenegro, Author

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor John Selker for all his scientific insight, support, and enthusiasm. I would also like to acknowledge James Wagner for all the ideas and support he offered and for his belief in the project. I would also like to thank Christoph Thomas for allowing me to deploy HumiSense on his study site off of Highway 34 near OSU's Botany & Plant Pathology Experimental Farm. I would also like to thank my friends and lab mates Aida Arik, Tara O'Donnell, Daniel Bailey, and Ryan Stewart for their encouragement during the duration of this study. And last, but definitely not least, I would like to acknowledge my husband Gregory Gutshall for all his support, understanding, and love.

TABLE OF CONTENTS

Page

LIST OF FIGURES

HIGH FREQUENCY WATER VAPOR DENSITY MEASUREMENTS USING THE BEAT FREQUENCY METHOD

1 Introduction

Evapotranspiration and precipitation constitute the largest two terms in many water budgets, especially in irrigated environments like produce farms where not much runoff or changes in soil moisture or surface water are expected. This makes measuring both precipitation and evapotranspiration crucial, if we want our water budgets to be accurate. With the human population now exceeding7 billion (Roberts *2011*), accurate budgeting of water for farming and agriculture is necessary for sustainable food production sufficient to feed the expanding population. Currently, rain gauges and weighing lysimeters are the most accurate methods of measuring rainfall and evapotranspiration, respectfully. This is in part because rain gauges and weighing lysimeters are accurate, but lysimeters are very expensive and so, not accessible to many who need the data. Rain gauges and lysimeters also require periodic maintenance making them difficult for long term unattended use.

The Eddy Covariance method is another technique used to measure evapotranspiration based upon measurement of the covariance of vertical air movement and water vapor. The Eddy Correlation method is not in widespread use in agriculture, in part due to the lack of water vapor sensors that are inexpensive, reliable, and can achieve a fast response time (~20Hz required). This need gave rise to the idea for this project. If a reliable and inexpensive vapor density sensor can be developed, inexpensive ultrasonic anemometers can be coupled with the vapor density sensors to utilize the

Eddy Covariance technique and water vapor turbulent flux from crops can be quantified.

The water vapor density sensor undertaken was implemented using the beat frequency method from Pungs and Preuner (Tregidga *1940*), where rapid changes in water vapor content will be obtained from changes on the dielectric constant of an open air capacitor. The beat frequency method has been used substantially in the past in refractometers to measure changes in the refractive index of the atmosphere (Crain *1950*), in refractometers to measure changes in the refractive index of different gases (Hay, Martin, and Turner *1961*), in microwave hygrometers to measure changes in the water vapor pressure in air (Stokesberry and Hasegawa *1976*, Hasegawa and Stokesberry *1975*, Sargent *1959*), in hygrometers to measure changes in the dielectric constant (Crain *1948)* and dipole moment of different gases (Jelatis *1948*), in hygrometers to measure changes in the dielectric constant of water vapor (Tregidga *1940*) , and in hygrometers to measure changes in the dielectric constant of air (Webb and Neugebauer *1954*). The reported sensitivities of the different devices employing the beat frequency method ranged from 200 ppm (Tregidga *1940*), 56.1 ppm (Sargent *1959*), 10.7 to 21.4 ppm (Crain *1950*), to 0.4 ppm (Jelatis *1948*).

When using the beat frequency method, there is always a sensing circuit and a reference circuit. The signals of both circuits are mixed (frequency heterodyning), so that small changes in the sensing circuit can be noted more easily. Our reference

circuit was a 16 MHz crystal Colpitts oscillator because of its simplicity, with 16 MHz being the maximum frequency for which inexpensive chip microcontrollers are available. Our sensing circuit was a LC Clapp oscillator with a frequency near 16 MHz. The main sensing device in the sensing circuit was an open air capacitor (one of the LC oscillator constituent parts), who's capacitance changed directly as the dielectric constant of the air varied. This varying capacitance made the LC oscillator a Variable Frequency Oscillator (VFO). The water vapor content of air was the dielectric signal of interest, with temperature and atmospheric pressure also affecting the dielectric constant of air, and thus the capacitance of the open air capacitor.

Changes in the dielectric-effected capacitance due to three environmental signals were measured using a beat frequency method, where an electronic mixer combined the signal from the LC oscillator with the reference signal from a crystal oscillator to obtain a compound output signal (frequency heterodyning). The compound output signal had two constituent parts, first the high frequency signal made of the addition of the LC oscillator frequency and the reference frequency, and second the low frequency signal made of the subtraction of the LC oscillator frequency and the reference frequency. Since both, the LC frequency and the reference frequency were chosen to be close in value (near 16 MHz), the high frequency components of the signal were removed with a low-pass filter (LPF), and only the low frequency signal remained for processing, which then can be completed with an inexpensive low sample rate microcontroller. A low pass filter was used to separate the high frequency component

from the low frequency component of the output signal. The output of the heterodyne signal was processed by an Atmel Mega168 microcontroller unit (MCU). The entire circuit and system is referred to as the HumiSense. The concept here is that HumiSense would provide reliable water vapor density sensor for high frequency fluctuations as required for eddy covariance computations, which would then be coupled with water vapor density low frequency sensor to have a full spectrum water vapor density signal.

2 Literature Review of the Beat Frequency Method in Hygrometers and Refractometers

Tregidga (1940) constructed a “heterodyne beat apparatus” for measuring the dielectric constant of water vapor. In his apparatus there were two oscillators, a test oscillator and the reference oscillator. The test oscillator was a Hartley oscillator with three parallel plate capacitors connected in parallel. The reference oscillator was a 10.58 MHz crystal oscillator. The frequency of the crystal was quadrupled via a frequency multiplier circuit to reach their desired frequency of 42 MHz. Changes in water vapor concentration were imposed on what was referred to as the gas capacitor; the other two capacitors were the measuring capacitor and the tuning capacitor. The frequency would change when the gas capacitor was subjected to changes in water vapor concentration and the changes in frequency would then be counterbalanced by change in capacitance of the measuring capacitor. The measuring capacitor had a range of capacitance between 2.0 and 2.1 pF. The changes in capacitance of the measuring capacitor were done by a National Vernier drive (the National Company, Malden, MA, USA). The measuring resolution was of 1 part in 5000 and the capacitor was calibrated to the selected frequency of 42 MHz. The gas capacitor was a parallel plate capacitor made up of cone-shaped Invar plates with a diameter of 3.81 cm separated by 0.102 cm. Ultra-low thermal expasion Invar plates were used instead of brass plates to avoid effects on plate size due to temperature fluctuations. To remove effects of stray capacitances on the oscillator frequency, air was used as the calibrator

gas. To reduce drift effects, the measurements done always alternated between water vapor and vacuum. The dielectric constant was measured at 1 atm.for the pressure range of 0 to 0.86 atm for the temperatures of 71.9°C, 99.8°C, and 147°C. From his results, Tregigda agreed that the quantity ($\varepsilon - 1$), where ε is the dielectric constant of a non-polar gas, varied linearly with the density of the nonpolar gas (air was treated as a nonpolar gas in frequencies lower than 42 MHz at the time; currently air is considered a mix of polar and nonpolar components). In his results, he concluded that the dielectric constant of water vapor at 1atm was approximately 1.0071, 1.0060, and 1.00475 for the temperatures 71.9°C, 99.8°C, and 147°C.

Jelatis (1948) did a study using the beat frequency method to measure the dielectric constant of several gases, among them helium, argon, oxygen, and neon. In his study, the test oscillator was a Colpitts oscillator with three capacitors, one of which was the gas capacitor. The reference oscillator was a 999 kHz quartz oscillator. The test oscillator had a frequency of 1 MHz. When measuring changes in the gas capacitor, Jelatis' circuit had a maximum sensitivity of 0.0002 pF, which translated to changes of 0.4 ppm ($\Delta C/C_G = 0.0002pF/500pF$). During the same year as Jelatis, Crain (1948) was interested in measuring fluctuations in the dielectric constant of dry air, oxygen, water vapor, chloroform, acetone, ethyl alcohol, and ethyl chloride at a frequency of 9,340 MHz. He used the beat frequency method employing a reference oscillator and a test oscillator. In his 1948 study, Crain did not document his sensitivity, but in his

1950 study on the refractive index of the atmosphere, he calculated a sensitivity range of gas refractive index of 10.7 ppm to 21.4 ppm.

A few years later, Webb and Neugebauer (1954) performed a study on body heat loss via expired air in a range of different temperatures. They measured body heat loss by recording the dielectric constant of the humid air expired by the subjects from ambient temperature to -42°C. Their device was designed following the beat frequency method. The final sensing capacitor that suited their conditions of small size, maximal plate area, and minimal temperature response was a coaxial capacitor made of thin Invar cylindrical plates covered in Plexiglas for thermal isolation. Their sensing capacitor had a capacitance of 5 pF, and their total capacitance was 21 pF. As part of their capacitor network, they had a coarse tuning capacitor and a fine tuning capacitor. They recorded a difference in frequency of 2.5 kHz when comparing dry air to moist air at 37°C. The sensitivity of their circuit is not documented in their publication.

Sargent (1959) constructed a microwave refractometer to measure the water vapor pressure of a moving parcel of air. His refractometer was a similar device to the one constructedby Crain (1948) with a dry air reference cavity connected to a reference resonator and a test cavity connected to the test resonator. The instrument recorded the changes in refractive index of air due to fluctuations in relative humidity. The refractometer was calibrated using pure dry gases of known refractive indexes. His

refractometer had a sensitivity of 56.1 ppm. In 1961, Hay *et al.* also designed and built a gas refractometer using the beat frequency method as their sensing mechanism. The test oscillator was a Clapp (a.k.a. series-tuned Colpitts) oscillator with a resonant frequency of 10 MHz. Their resonator had an inductor of 6.1μH and a total capacitance of about 50 pF. The refractometer circuit was contained inside a Bakelite box for insulation. The sensing capacitor was a parallel plate capacitor made up by twelve plates 1.600 cm by 0.152 cm separated by 0.635 cm to generate a total capacitance of 50 pF. Invar steel plates and fused quartz spacers (both with very low thermal expansion) were used in between plates without adding a significant temperature coefficient to the capacitor. Instead of having a dedicated reference oscillator, a precision switch was used to toggle between the sensing capacitor and a reference capacitor. The reference capacitor had a capacitance range between 51 and 62 pF that was not affected by the air's dielectric constant. A frequency meter was used to translate the frequency signal to DC voltage, which was at the end stored by a recorder pen. The sensitivity of this system was not noted in the publication.

Lastly, Stokesberry and Hasegawa created an automatic digital microwave hygrometer and they described it in two publications (Hasegawa and Stokesberry *1975,* Stokesberry and Hasegawa *1976*). The hygrometer was constructed following the two cavity (beat method) refractometer idea where each cavity resonates at a frequency that depends on the cavity's shape and on temperature, pressure, and the dielectric constant of the gas in the cavity. The hygrometer was designed to measure the

humidity of atmospheric air in the water vapor range between 3 and 7400 Pa. The reference cavity and the test cavity were both kept in at the same temperature and same total pressure by means of a thermostatically controlled oven. The hygrometer was made up by the two oscillator or cavities (the test cavity and the reference cavity), a balanced mixer circuit that removed the high frequency data leaving only the difference in frequency between the two cavities, and a commercial frequency counter. Each count represented a difference of 70 Hz between the test and the reference cavities. The difference between their 1976 study and their 1975 study was the removal of mechanical parts that were either sensitive to mechanical wear or sensitive to mechanical vibrations.

Our objective was to develop a device following the model of those discussed above but using contemporary integrated circuits and microprocessing to make a low-cost and accurate device to measure the fluctuations of air humidity with temporal resolution between 0.1 and 20 Hz.

3 Materials and Methods

3.1Measuring water vapor density with a capacitive sensor

The following concepts are crucial to understanding the relationship between the dielectric constant of a capacitive sensor to the ambient water vapor density. The equations for vapor pressure, relative humidity, and vapor density were obtained from Kerr's book *Propagation of Short Radio Waves* from 1951. The relationship between vapor pressure and relative humidity are essential for understanding many of the derivations in the methods chapter:

Vapor Pressure (*e*)

The *vapor pressure* equation approximates the relationship between pressure, density, and temperature of water vapor in the atmosphere. This relationship is almost nearly independent of the dry-air components present. In unit of Pascals, water vapor pressure in air is given by:

$$e = \frac{R}{m_v}\rho_v T \tag{1}$$

where R is the universal gas constant with a value of 8.314 J mol^{-1} K^{-1}, m_v is the molecular weight of water with an approximate value of 18.015g mol^{-1}, ρ_v is the water vapor density or absolute humidity with units of kg m^{-3} and T is the absolute temperature in K.

Relative Humidity (*f*)

Relative humidity represents the relation between vapor density (ρ_v) and saturation vapor density (ρ_{vs}) at the same temperature and pressure. Since water vapor is treated as an ideal gas, relative humidity is also equivalent to the ratio between vapor pressure (e) and saturation vapor pressure (e_s):

$$f \equiv \frac{\rho_v}{\rho_{vs}} = \frac{e}{e_s} \tag{2}$$

Vapor Density (ρ_v)

Vapor density is the mass of water vapor (kg) found in a specific volume of air (m^3). Combining equations (1) and (2), we get:

$$\rho_v = \frac{e * m_v}{R * T} = \frac{e * 10^6}{R_v * T} \tag{3}$$

where vapor density is in units of kg $m^{-3}$and R_v is the specific gas constant for water vapor, 461.495 J kg^{-1} K^{-1}. R_v is equal to R*10^6*m_v.

Saturated Vapor Pressure (e_S)

Saturated vapor pressure is accurately determined using the Clausius-Clapeyron equation for water vapor (Salzman 2001):

$$\frac{de_s}{dT} = \frac{L_v e_s}{R_v T^2}$$

where e_s is saturated vapor pressure in Pa, L_v is the specific latent heat of vaporization which is 2.5 MJ kg^{-1}, T is the absolute temperature in K, and R_v is the water vapor gas constant which is 461.5 J kg^{-1} K^{-1}.

Saturated vapor pressure can also be approximated usingthe polynomial equation derived by Lowe (1977) in his publication entitled "An Approximating Polynomial for the Computation of Saturation Vapor Pressure". The equation he presents is the following:

$$e_s = a_0 + a_1T + a_2T^2 + a_3T^3 + a_4T^4 + a_5T^5 + a_6T^6 \quad (4)$$

where, e_s is the saturated vapor pressure in units of mb and T is the temperature in K. The coefficients are,

- a_o = 6984.505294
- a_1 = -188.9039310
- a_2 =2.133357675
- a_3 = -1.288580973E-02
- a_4 = 4.393587233E-05
- a_5 = -8.023923082E-08
- a_6 = 6.136820929E-11.

This approximation is valid for the temperature range of -50°C to 50°C for water reference and -50°C to 0°C for ice references. The errors for the approximation are below 1% except for at -45°C to -50°C were the error is approximately 1%. No restrictions on atmospheric pressure are reported.

Specific humidity (q)

Specific humidity is the ration of mass of water vapor (kg) to mass of moist air (kg) (Brutsaert 1982):

$$q = \frac{m_v}{m} = \frac{\rho_v}{\rho} \quad (5)$$

where m_v is the mass of water vapor in kg and m is the mass of moist air ($m = m_v + m_d$ where m_d is the mass of dry air) in kg, ρ_v is the density of water vapor in kg m^{-3}, ρ and is the density of moist air kg m^{-3} ($\rho = \rho_v + \rho_d$ where ρ_d is the density of dry air).

Mixing ration (m)

Mixing ration is the ration of mass of water vapor (kg) to mass of dry air (kg) (Brutsaert 1982):

$$m = \frac{\rho_v}{\rho_d} \quad (6)$$

where ρ_v is the density of water vapor in kg m^{-3} and ρ_d is the density of dry air kg m^{-3}. Water vapor content of air can be expressed in terms of mixing ratio.

3.2 Dielectric Constant of Water Vapor as a Polar Gas

By assuming the ideal gas law, the equation for dielectric constant of a non-polar gas, ε, can be described by (Bean 1962):

$$\varepsilon - 1 = K_1' * \frac{P}{T} \tag{7}$$

where, P is absolute pressure in units of mb, T is absolute temperature in K, and K_1' is a constant.

For polar gases, assuming the ideal gas law, the relation changes to the following:

$$\varepsilon - 1 = K_2' * \frac{P}{T} * \left(A + \frac{B}{T}\right) \tag{8}$$

where, K_2' , A, and B are constants.

Air is considered to be a mixture of non-polar gases and water vapor (Bean 1962). The non-polar components of air (oxygen, nitrogen, and carbon dioxide) have little effect on the dielectric constant of air. Water vapor, due to its polar nature, can raise the dielectric constant of air when its concentration increases. According to Dalton's law of partial pressures, to obtain the value of the dielectric constant of air, the effects of both polar and non-polar gases can be summed together. Thus, the equation for dielectric constant of air will look as follows:

$$\varepsilon - 1 = \sum_{i=1}^{j} K_{1i}' * \frac{P_i}{T} + \sum_{l=1}^{m} K_{2l}' * \frac{P_l}{T} * \left(A_l + \frac{B_l}{T}\right) \tag{9}$$

where, the first term of the sum is made up of the non-polar gases and the second term of the polar gases (i.e. water vapor). Hence, the equation for dielectric constant of air will have the following behavior:

$$\varepsilon - 1 = \alpha * K_1' * \frac{P_d}{T} + \beta * K_2' * \frac{e}{T} * \left(A + \frac{B}{T}\right) \quad (10)$$

where, α is the proportion of dry components, β is the proportion of water vapor, T is temperature, P_d is the pressure of dry air (corresponding mostly to nitrogen, oxygen, argon, and CO_2) and e is the partial pressure of water vapor (i.e. vapor pressure). K_1' and K_2' are constants.

Englund et al. (1935) deduced the values of K_1', K_2', A, and B and published the following equation for the dielectric constant of moist air:

$$\varepsilon - 1 = \left[(\alpha * 211) + \left(\beta * 182\left(1 + \frac{5{,}582}{T}\right)\right)\right] * 10^{-8} * \frac{P_t}{T} \quad (11)$$

where, P_t is the total pressure of the moist air, α is the percentage of dry gases in the air and β is the percentage of water vapor in the air.

We can then replace, α with its ratio metric equivalent $(100 - \beta)$ giving us:

$$\varepsilon - 1 = \left[211 + \beta\left(\frac{10{,}159.24}{T} - 0.29\right)\right] * 10^{-6} * \frac{P_t}{T} \quad (12)$$

If we do as Lea (1945), and express β in terms of relative humidity by forcing the vapor pressure to equal the pressure of dry air $P_d = P_t - e$ (page 265), the volume of the water vapor would then from the Combined Gas Law, become:

$$e * V_t = P_d * V_{wv_c} \tag{13}$$

So,

$$V_{wv_c} = V * \frac{e}{P_d} = V * \frac{e}{P_t - e} \tag{14}$$

Then β becomes:

$$\beta = \frac{\left(V * \frac{e}{P_t - e}\right)}{V + \left(V * \frac{e}{P_t - e}\right)} * 100 = \frac{e}{P_t} * 100 \tag{15}$$

And if we recall, relative humidity (f) has the following relation:

$$f(\%) = \frac{e}{e_s} * 100 \tag{16}$$

So, then β can be converted to:

$$\beta = \frac{e_s * f}{P_t} \tag{17}$$

And then, the dielectric constant equation becomes:

$$\varepsilon - 1 = \left[211 + \left(\frac{e_s * f}{P_t}\right)\left(\frac{10{,}159.24}{T} - 0.29\right)\right] * 10^{-6} * \frac{P_t}{T}$$

$$= \left[\frac{211P_t}{T} + \left(\frac{e_s * f}{T}\right)\left(\frac{10{,}159.24}{T} - 0.29\right)\right] * 10^{-6}$$

$$= \frac{211}{T}\left[P_t + \left(\frac{48.148}{T} - 0.0014\right) * (e_s * f)\right] * 10^{-6} \qquad (18)$$

A very close approximation of equation (18) was used by Bertone et al. (1991) to calculate the dielectric constant of moist air. Equation (18) can be modified so that instead of having relative humidity (*f*) and saturated water vapor pressure (e_s) as inputs, we can have instead water vapor density. The equation would then become:

$$\varepsilon_{r_{moist\ air}} = 1 + \frac{1.583}{T}\left[P_t + \left(\frac{48.148}{T} - 0.0014\right) * e\right] * 10^{-6}$$

$$= 1 + \frac{1.583}{T}\left[P_t + (48.148 - 0.0014T)\frac{R * \rho_v}{m_v}\right] 10^{-6} \qquad (19)$$

where, R is the universal gas constant in $J\ mol^{-1}\ K^{-1}$, m_v is the molecular weight of water in units of *kg/mol*, ρ_v is the water vapor density or absolute humidity in kg/m^3, T is the absolute temperature in units of *K*, and P_t is pressure of moist air in units of *Pa*.

If the value for the high frequency dielectric constant of air is obtained experimentally with HumiSense, then the high frequency water vapor density can be derived from equation (19), assuming that P_T and T have been independently calculated and the saturated vapor pressure has been estimated using equation (4).

3.3 Modeling the changes in Capacitance due to Water Vapor, Temperature, and Atmospheric Pressure

Equation (18) can be used to model the effects of relative humidity, atmospheric temperature, and atmospheric pressure on the dielectric constant of moist air. Because the value of 0.0014T in equation (18) is less than one tenth of 48.148 for the values of temperature in which we are interested, specifically for the purposes of sensitivity analysis the equation can be simplified to:

$$\varepsilon_{r_{moist\ air}} = 1 + \frac{211}{T}\left[P_t + \frac{48.148}{T} * e_s * f\right] * 10^{-6} \tag{20}$$

We can rearrange equation (20) to make it easier to see the effects of humidity, temperature, and atmospheric pressure. By expanding equation (20), we get:

$$\varepsilon_{r_{moist\ air}} = 1 + \frac{211}{T} * P_t * 10^{-6} + \frac{211}{T} * \frac{48e_S}{T} * f * 10^{-6} \tag{21}$$

By inspection of equation (21) we can judge that a +1% change in RH would cause the dielectric constant to change by:

$$\begin{aligned}\Delta\varepsilon_{r_{1\%\Delta RH}} &= \left[1 + \frac{211}{T} * P_t * 10^{-6} + \frac{211}{T} * \frac{48e_S}{T} * (f+1) * 10^{-6}\right] \\ &\quad - \left[1 + \frac{211}{T} * P_t * 10^{-6} + \frac{211}{T} * \frac{48e_S}{T} * f * 10^{-6}\right] \\ &= \frac{211}{T} * \frac{48e_S}{T} * 10^{-6}\end{aligned}$$

$$= \frac{e_S * 1.013 * 10^{-2}}{T^2} \sim 1ppm$$

A change of +1 ºK for T≈300K would cause the dielectric constant to change by:

$$\Delta\varepsilon_{r_{1deg\Delta T}} = \left[1 + \frac{211}{(T+1)} * P_t * 10^{-6} + \frac{211}{(T+1)} * \frac{48e_S}{(T+1)} * f * 10^{-6}\right] - \left[1 + \frac{211}{T} * P_t * 10^{-6} + \frac{211}{T} * \frac{48e_S}{T} * f * 10^{-6}\right]$$

$$= \frac{-2.11 * 10^{-4} * P_t * T^2 - 2.11 * 10^{-4} * P_t * T - 30.38 * 10^{-3} * e_S * f}{T^4 + 2T^3 + T^2} \sim -2ppm$$

A change of +1 mmHg(i.e. 0.0013atm) in the atmospheric pressure for P≈1atm would cause the dielectric constant to change by:

$$\Delta\varepsilon_{r_{1mm\ Hg\Delta P}} = \left[1 + \frac{211}{T} * (P_t + 1) * 10^{-6} + \frac{211}{T} * \frac{48e_S}{T} * f * 10^{-6}\right] - \left[1 + \frac{211}{T} * P_t * 10^{-6} + \frac{211}{T} * \frac{48e_S}{T} * f * 10^{-6}\right]$$

$$= \frac{211 * 10^{-6}}{T} \sim 0.7ppm$$

So, a 1% increase in relative humidity can increase the dielectric constant of moist air by 1ppm, an increase of 1 ºK can decrease the dielectric constant of moist air by 2ppm, and a 1 mmHg increase in atmospheric pressure can cause a change of 1ppm in the dielectric constant of moist air. In the next section, the Eddy Covariance method will be introduced as an alternative to measuring humidity for high frequency needs.

3.4 An affordable device for measuring instantaneous specific humidity in an Eddy Covariance machine

The ultimate goal of this research would be to have HumiSense be the water vapor density sensor in an Eddy Covariance instrument. The eddy covariance method for measuring vertical turbulent fluxes was developed by Swinbank (1951). This method is based on the following relations for average vertical transfer of water vapor (*E*) and sensible heat (*H)*:

$$E = L\rho\,\overline{w'q'}, \tag{22}$$

and

$$H = C_p\rho\,\overline{w'T'}, \tag{23}$$

where *L* is the latent heat of vaporization of water, and C_p the specific heat of air at constant pressure, ρ' is the perturbation in air density, w' is the perturbation in vertical wind velocity , q' is the perturbation in specific humidity, and T' is the perturbation in air temperature. The line above signifies the cross-correlation of the parameters over a suitable averaging period usually between 15 and 30 minutes, and sometimes as high as 1 hour.

The relations for Eddy Covariance are derived following Reynolds Decomposition. In fluid dynamics and turbulence theory, the Reynolds Decomposition states that all variables in the system can be decomposed into mean and fluctuating parts (Alfonsi 2009). The fluctuating part is known as the perturbation in the variable. For example,

the average vertical transfer of water vapor (*E*) has vertical wind velocity and specific humidity. Each variable, vertical wind velocity and specific humidity, has a mean part and a fluctuating part:

$$w = \overline{w} + w'$$

and

$$q = \bar{q} + q'$$

When calculating *E* in Eddy Covariance, both mean and fluctuating parts interact, but only the interaction between fluctuating parts is relevant to Eddy Covariance, so the other terms are dropped.

Ultimately, it would be useful to have an affordable water vapor density sensor that was accurate and precise enough to be part of an eddy covariance system, especially since affordable ultrasonic anemometers can be used to measure the vertical wind velocity by measuring the shift in time of flight. If high frequency measurements for absolute temperature, absolute pressure, and vertical wind velocity were known, the values for all the parameters in equation (22) would be known except for the instantaneous specific humidity. HumiSense high frequency dielectric constant measurements could be used to obtain the instantaneous specific humidity (see equation 5) and ultimately knowing the values for all the parameters in equation (22), the vertical transfer of water vapor (*E)* could be calculated.

3.5 The Electrical characteristics of HumiSense

Figure 1 is a photograph of the HumiSense prototype made up of the sensing variable frequency oscillator, the reference oscillator, the Low Pass filter, the mixer unit, the microprocessing unit, the RS232 communication chip, and all the power circuitry needed.

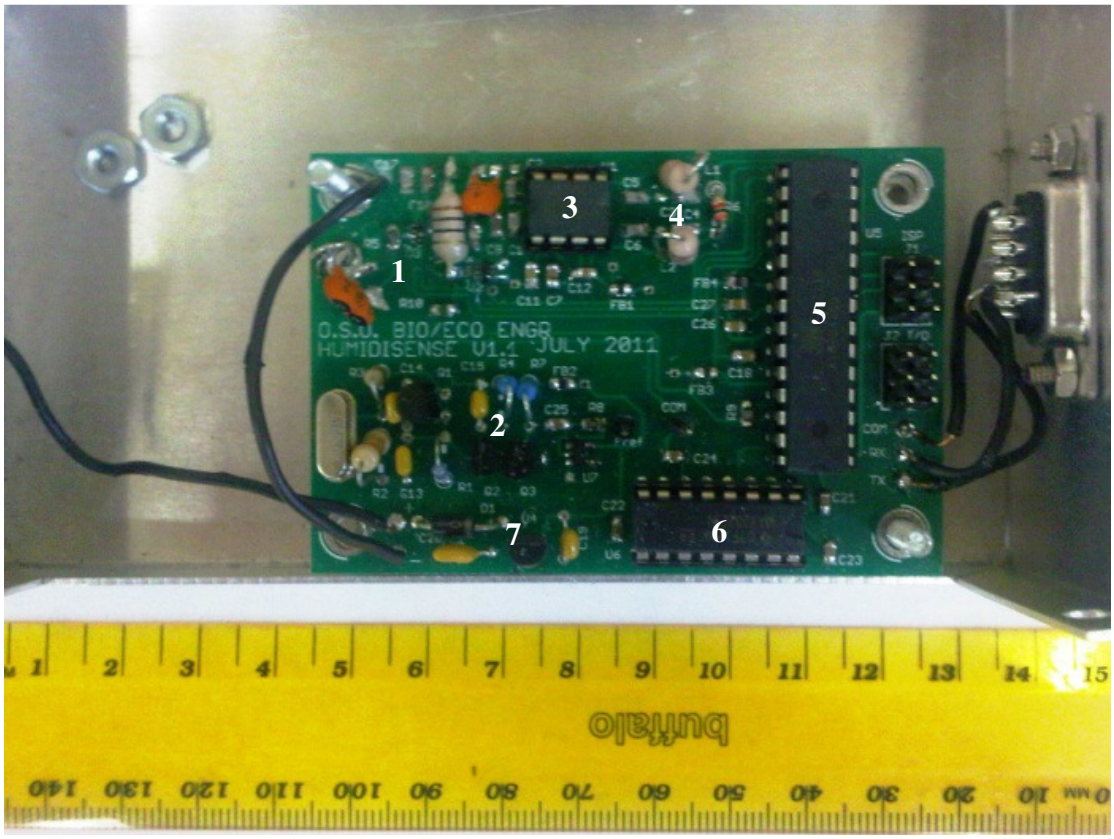


Figure 1. The HumiSense Main Board:*1* is the variable (a.k.a. sensing) oscillator; *2* is the reference oscillator; *3* is the mixer unit; *4* is the low pass filter; *5* is the microprocessing unit (MCU) or microcontroller; *6* is the RS232 communication chip (MAX232); and *7* is the voltage regulating circuitry.

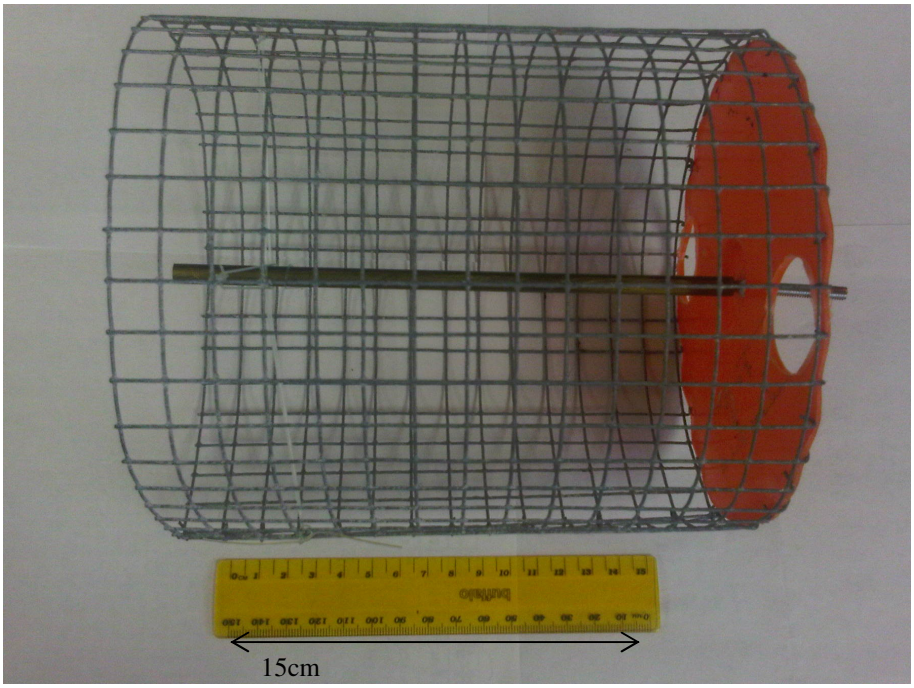


Figure 2. The HumiSense Sensing Capacitor.

3.5.1 The HumiSense System

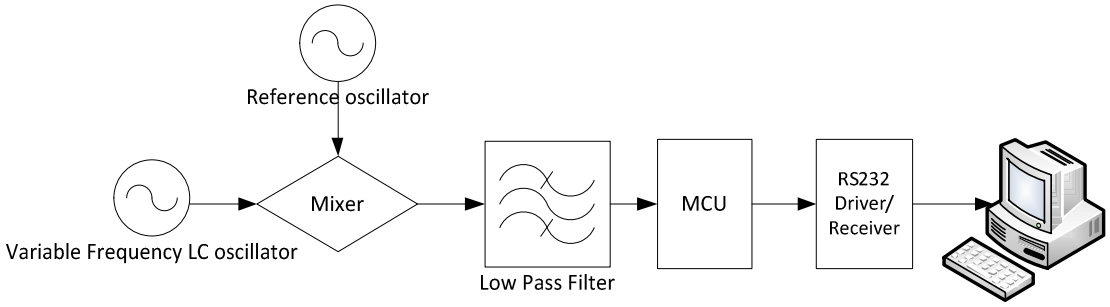


Figure 3. Flowchart of the HumiSense system.

The reference oscillator was a crystal oscillator with a resonant frequency of 16 MHz. For the HumiSense prototype, an Abracon 16 MHz crystal model ABL-16.000MHZ-B2 was selected. The Variable Frequency Oscillator was set at a frequency close to 16

MHz, either slightly higher or slightly lower than the reference oscillator. The Variable Frequency Oscillator will be explained in detail in the following sections of this chapter. The frequency mixer (or just mixer as in Figure 3) was a NPX mixer model SA602A. The SA602A is a low-power Very High Frequency double-balanced mixer that also includes an input amplifier, an on-board oscillator, and a voltage regulator.

The SA602A mixer was used to heterodyne the frequency of the reference oscillator and the frequency of the variable frequency oscillator. The heterodyne signal consists of two frequencies: a high frequency which is equal to the sum of the two input frequencies and a low frequency which is equal to the difference of the two input frequencies. The frequency of interest was the low frequency or the difference frequency because it allowed small changes in the variable frequency (one of the input frequencies) to be more easily read and also because it can more easily be handled by a low cost, low power counter. The mixer had two differential outputs, meaning that the signal from one output was inverted when compared to the signal of the other output. A Low Pass filter was connected to each of the mixer outputs. Both Low Pass Filters were the same; they were made up of two 270 pF capacitors and a 220 uH inductor giving rise to a 3 db cutoff frequency at about 460 kHz. To save on parts the ground connections of both filters were connected together.

A Low Pass Filter cutoff frequency of 460 kHz meant that all signals with a frequency of 460 kHz or higher would be attenuated. Each filter attenuated the sum frequency as well as any leakage signal from either oscillator, making the difference frequency easier to measure. Then, the microcontroller unit, an ATmega168 from Atmel (San Jose, CA, USA), was used to quantify and qualify the frequency shifts in the variable frequency oscillator with the controller's built-in differential comparator that provided a compatible input to the internal counters of the controller. Lastly, a Maxim (San Jose, CA, USA) MAX232 RS232 driver/receiver was used to deliver the frequency shifts to an external computer via serial communications.

3.5.2 The LC Oscillator

The resonator of a basic LC oscillator is made up of one capacitor and one inductor, as shown below:

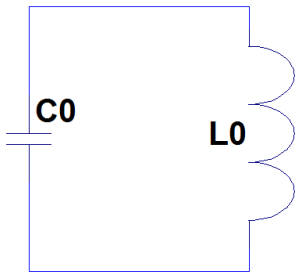


Figure 4. LC resonator

The oscillator oscillates only at the resonant frequency (Carr 1993), which is given by the following equation:

$$F_0 = \frac{1}{2\pi\sqrt{L_0 C_0}} \quad (24)$$

where, L_0 is in unit of Henries, C_0 is in units of Farads, and F_0 is in Hertz.

For HumiSense, the variable frequency or sensing oscillator was chosen to be a 16 MHz LC oscillator. The frequency of 16 MHz was selected because it was the highest frequency that would allow the use of the same signal to clock low cost, low power microcontrollers, leading to the highest counter resolution for either period or frequency measurement. We wanted the highest frequency possible because small changes in resonator capacitance result in large frequency changes, and since the heterodyne process results in a difference frequency, the large changes in frequency can easily be measured in the difference frequency. Also, with readily available inductors, a resonant circuit could be constructed with small but practical capacitances, leading to improved sensitivity, as will be shown later.

A 10uH inductor was chosen for the resonator circuit of the oscillator because an inductor of this inductance is readily available in small sizes and also because it results in relatively small resonating capacitances at 16 MHz. Having small capacitance helps improve sensitivity because the sensing capacitor is small in capacitance value due to inherent geometric and dielectric restrictions. Knowing F_0 is 16 MHz and L_0 is 10uH, from rearranging equation (24) C_0 becomes :

$$C_0 = \frac{1}{F^2\, 4\pi^2 L_0}$$

$$= \frac{1}{4\pi^2[(16.00MHz)^2 * (10.00uH)^2]}$$

$$= 9.89 * 10^{-12}F = 9.89pF$$

3.5.3 A Note on the Dielectric Constant of Air at 16 MHz

According to Hughes and Armstrong (1951), the dielectric constants of air are independent of frequency from DC to near the first absorption band that is above 30 GHz. Hence, the dielectric constant of air in the lower atmosphere should not be affected by oscillator and other electronic devices at frequencies below 30 GHz.

3.5.4 Frequency Sensitivity due to Capacitance Deviations in the Sensing oscillator

To understand the sensitivity of the resonant frequency to the changes in capacitance, let's first assume that C_0 changes by ΔC, equation (23) becomes:

$$F_0{}' = \frac{1}{2\pi\sqrt{L_0(C_0 + \Delta C)}}$$

$$= \frac{1}{2\pi\sqrt{L_0 C_0\left(1 + \frac{\Delta C}{C_0}\right)}}$$

$$= \frac{1}{2\pi\sqrt{L_0 C_0}} * \frac{1}{\sqrt{\left(1 + \frac{\Delta C}{C_0}\right)}}$$

$$= F_0 * \frac{1}{\sqrt{\left(1 + \frac{\Delta C}{C_0}\right)}}$$

Since $\frac{\Delta C}{C_0} \ll 1$, the value of $1 + \frac{\Delta C}{C_0}$ can be approximated using the Binomial Series. The Binomial Series is the Taylor Series expansion of the function given by $f(x) = (1 + x)^{\alpha}$ about $x = 0$, where α is an arbitrary complex number and in our case it is equal to 2.

So,

$$\cong F_0 * \frac{1}{1 + \frac{\Delta C}{2 * C_0}}$$

$$\cong F_0 * 1 - \frac{\Delta C}{2 * C_0}$$

Then,

$$\frac{F_0{}'}{F_0} = 1 - \frac{\Delta C}{2 * C_0} \tag{25}$$

Due to the difference in capacitance between F_0 and F_0' the following is known:

$$F_0{}' = F_0 + \Delta F$$

$$= F_0 \left(1 + \frac{\Delta F}{F_0}\right) \tag{26}$$

Then, by relating equations 25 and 26, we get that:

$$\frac{F_0{}'}{F_0} = \frac{F_0 \left(1 + \frac{\Delta F}{F_0}\right)}{F_0}$$

$$= 1 + \frac{\Delta F}{F_0}$$

$$\cong 1 - \frac{\Delta C}{2 * C_0}$$

and

$$\frac{\Delta F}{F_0} \cong -\frac{\Delta C}{2 * C_0} \tag{27}$$

This means that a fractional change in C_0 will cause half that fractional change in F_0. For example, assume that C_0 changes by 3.43 aF (10^{-18} Farads) which is 1ppm of the sensing capacitor (see the next section) used for this project (1ppm corresponds to a change of 1% relative humidity according to section 3.3 of this document), F_0 will approximately change by:

$$\Delta F \cong -F_0 * \frac{\Delta C}{2 * C_0}$$

$$= -16.00MHz * \frac{3.43aF}{2 * 9.89pF}$$

$$= -2.77Hz$$

Changes of at least 2.77 Hz will need to be measurable by our data processing algorithm in order to see changes of 1ppm in our sensing capacitor.

3.5.5 Sensing Capacitor (C_{sense})

The sensing capacitor (C_{sense}) was modeled and made as a coaxial or cylindrical capacitor. The coaxial capacitor follows this capacitance equation (Shen and Kong 2007; Lorrain and Corson 1990):

$$C_{Coaxial} = \frac{2\pi\varepsilon_0\varepsilon_r d}{\ln\left(\frac{b}{a}\right)} \tag{28}$$

where, ε_0 is the dielectric constant of free space ($\varepsilon_0 \approx 8.854 \times 10^{-12}$ F/m), ε_r is the dielectric constant of the medium, d is the length of the capacitors, a is the diameter of the inner electrode, and b is the diameter of the outer electrode.

The radial dimensions of the capacitor are b and a, which represent the outer and inner dimensions respectfully. d is the length dimension of the cylinder. The sensing capacitor was designed keeping in mind that it needed to be portable, small, and it needed to allow air to flow through it. Given that the dielectric constant of free space is in the picofarad per meter range, we chose dimensions that would keep the sensor portable and small but that would also allow the sensing capacitor to have a "considerable" sample of air contained and be significantly affected by changes in the air dielectric constant, in order to be visible in the resulting resonant frequency. The sensor's length should match the day time high frequency turbulence path lengths. Following the Taylor hypothesis (Taylor 1938), assuming an average wind speed of

2m/s and with a frequency range of interest between 10 and 20 Hz, we find that the day time high frequency turbulence path lengths are between 20 and 10 cm.

With these considerations in mind, the sensor capacitor was designed and built with the following dimensions:

- Inner electrode diameter = a = 0.63 cm
- Outer electrode diameter = b = 11.7 cm
- Length = d = 18 cm

Air must have the ability to circulate through the capacitor. The outer electrode was made of galvanized steel wire mesh with square opening of 1.25 cm. The inner electrode is made of 6.33mm copper wire. The outer electrode was connected to ground while the center electrode was connected to the capacitor network. The capacitor's center electrode plugs directly into the circuit board. The outer electrode of the capacitor was connected to the circuit board ground with a 2" long wire adding a minor amount (~1 nH) of inductance in series to the sensor.

According to equation (23), the capacitance is linearly related to the medium's dielectric constant (ε_r). To calculate ε_r, it is necessary to know the temperature, the density of saturated water vapor, the density of air, the relative humidity, and the volume inside the capacitor. The volume inside a coaxial capacitor corresponds to the volume inside the larger cylinder minus the volume inside the smaller cylinder:

$$V_{cc} = \pi b^2 d - \pi a^2 d \tag{29}$$

The approximate value of the dielectric constant of air (ε_r) at 1 atm is 1.00054 (Kaye and Laby 1995). Employing the dimensions of the unit, the dielectric constant of free space ($\varepsilon_0 = 8.854 \times 10^{-12}$ F/m) and the approximate expected capacitance of the sensing coaxial capacitor was:

$$C_{sensor} = \frac{2\pi\varepsilon_r\varepsilon_0 L}{\ln\left(\frac{b}{a}\right)} \tag{30}$$

$$= \frac{2\pi * 1.00054 * 8.854 * \frac{10^{-12}F}{m} * 0.18m}{\ln\left(\frac{11.7cm}{0.63cm}\right)}$$

$$= 3.43\ pF$$

3.5.6 LC Oscillator Topology

Three oscillator topologies were considered for this study: Hartley, Colpitts, and Clapp. The Hartley oscillator was not selected because it requires a tapped inductor which makes this topology relatively expensive and hard to replicate. The Colpitts and Clapp topologies are preferred over the Hartley topology because of their simplicity and high frequency stability (Clapp *1948*). The Clapp topology is just a modified Colpitts oscillator. The difference between the Clapp and the Colpitts oscillators is in the connection between the capacitors on the sensing side of the circuit

(represented by $C_{sensing}$ in Figures 5a, 5b, and 5c) and the feedback capacitors (represented by C_1 and C_2 in Figures 5a and 5b) .

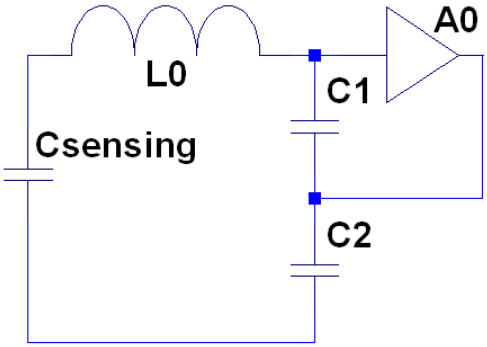


Figure 5a

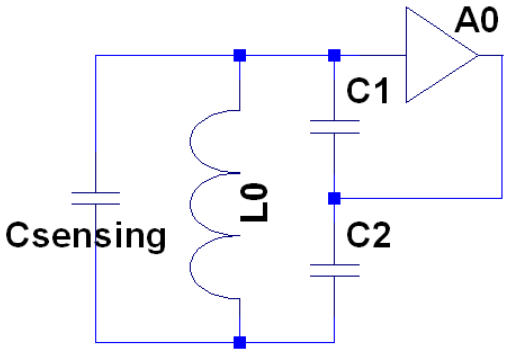


Figure 5b

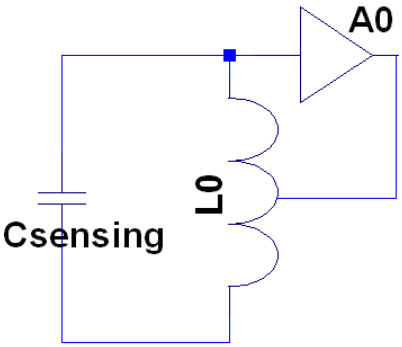


Figure 5c

Figure 5. (a) The schematic of a basic Clapp oscillator; (b) schematic of a basic Colpitts oscillator; (c)the basic schematic of the Hartley oscillator. The arrow in (c) indicates that the inductor is a tapped inductor. A_0 represents the active component i.e. the bipolar transistor in the three oscillators.

In the Colpitts oscillator, the capacitor network on the sensor side is represented by $C_{sensing}$ (see Figure 5b) and it is connected in parallel with the feedback capacitor network (C_{fb}), where:

$$C_{fb} = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} \tag{31}$$

In the Clapp oscillator, the capacitor network on the sensor side is in series with C_{fb}.

For both the Colpitts and the Clapp oscillators, the equation for the resonant frequency in Hertz is the following (same as equation 20):

$$F_0 = \frac{1}{2\pi\sqrt{L_0 C_0}}$$

In the Clapp topology, C_0 is calculated as follows:

$$C_0 = \frac{1}{\left(\frac{1}{C_{sensing}} + \frac{1}{C_{fb}}\right)}$$

$$= \frac{1}{\left(\frac{C_{sensing} + C_{fb}}{C_{fb} * C_{sensing}}\right)}$$

$$= \frac{C_{fb} * C_{sensing}}{C_{sensing} + C_{fb}}$$

$$= \frac{C_{sensing}}{1 + \frac{C_{fb}}{C_{sensing}}} \quad (32)$$

In the Colpitts topology, C_0 is calculated as follows:

$$C_0 = C_{sensing} + C_{fb} \quad (33)$$

When $C_{sensing}$ is substantially smaller than C_{fb}, a change in $C_{sensing}$ will cause an observable change in F_0 in the Clapp topology, while it will be barely noticeable in the Colpitts topology. This is due to the way C_0 is calculated in both oscillator topologies (equations 32 and 33). Given that in our design, $C_{sensing}$ is only about one forth of C_{fb}, we chose the Clapp topology. Choosing the Clapp topology when $C_{fb} >> C_{sensing}$ means that $\Delta C_0 \approx \Delta C_{sensing}$, and then $\Delta F_0 \approx 1/[2\pi\sqrt{(L*\Delta C_{sensing})}]$.

3.5.7 Circuit design

When energy is applied to the capacitor in an LC oscillator, the capacitor discharges the energy to the inductor as electric current and then the inductor builds up a magnetic field due to the current passing through it. Eventually, the capacitor discharges completely, but the current keeps flowing due to the magnetic field around the inductor. The current then charges the capacitor again in reverse and the cycle repeats. In theory, LC oscillators should oscillate forever after energy has been applied to them. In practice, the oscillation is damped with time because of resistances that are inherently present in the circuit (e.g. wire resistance, component resistance). To keep an oscillator running, the oscillator can be connected to an active component that introduces a positive feedback into the circuit. The positive feedback allows the oscillator to run "forever" by combining a portion of the output in phase with the input. To add positive feedback to our oscillator, we connected C_{fb} to a transistor inside the NXP SA602 Double-balanced mixer and oscillator. The SA602 is the part that performed the frequency heterodyning of the variable frequency oscillator and the reference oscillator.

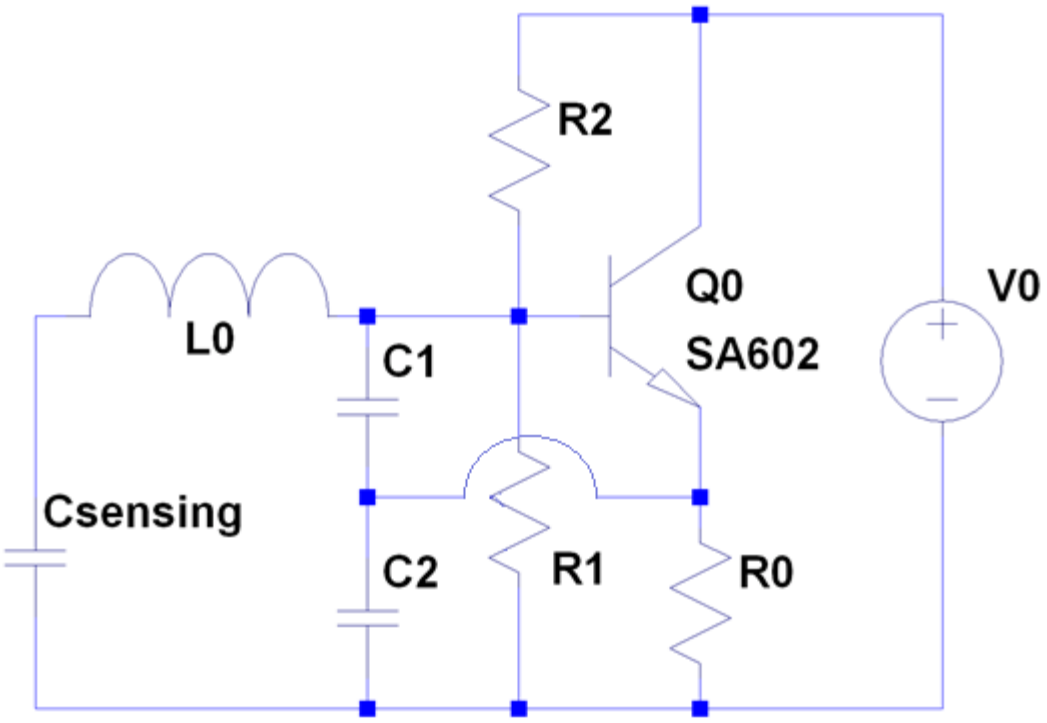


Figure 6. Schematic of the Clapp oscillator with the active component (SA602 transistor). The Cfb seen in Figure 5(a) has been replaced by two capacitances to create the positive feedback loop. The voltage supply V0 provides the energy for the oscillator to run. Resistor R0 controls the current flowing through the transistor, and resistors R1 and R2 bias the transistor, so that operates correctly.

Based on the value of the sensing capacitor and the oscillator topology, the values of the other capacitors in the circuit may be computed. As Figure 6 shows, to create the positive feedback with a bipolar junction transistor, C_{fb} must be replaced by two capacitors in series. The value of these two capacitors (C1 and C2) is crucial for the correct operation of the positive feedback. In the final operating circuit, C_{fb} will be made up of two equal value feedback capacitors (C1 and C2 in Figure 8), a fixed capacitor (C3 in Figure 8), and a Maxim MAX1474 digital trimmer (Tri1 in Figure 8) connected in series with a fixed capacitor (C4 in Figure 8). The feedback capacitors

will be connected to the active circuit and will provide the energy necessary for the circuit to oscillate. For optimum oscillator operation, we have learned from practice that the feedback capacitors need to be in the range of 27pF to 47pF each. Given that these feedback capacitors are connected in series, in combination their optimum value range is from 13.5pF to 23.5pF(almost ten times $C_{SENSING}$).

In the circuit employed, the two feedback capacitors were connected in series (C1 and C2 in Figure 8), the Maxim MAX1474 digital trimmer Tri1 and the fixed capacitor C4 were also connected in series (Figure 8), and together both pairs were connected in parallel to the fixed capacitor C3 (Figure 8). The feedback capacitors, the fixed capacitors, and the digital trimmer make up what is shown as C_{fb} in Figure 5a. $C_{sensing}$ is made up of four capacitors: the Maxim MAX1474 digital trimmer Tri2 in series with the fixed capacitor C5 (Figure 8), together connected in parallel to the coaxial sensing capacitor Csensor (Figure 8), and the fixed capacitor C6 (Figure 8). There were also parasitic capacitances (a.k.a. stray capacitances) present in the circuit, which are shown in Figure 7 as Cp1 and Cp2. The values of these capacitances are unknown, but their presence needs to be kept in mind to understand the performance (Figure 7). Parasitic capacitances are unwanted but unavoidable capacitances that show up due to the proximity of electronic parts. These capacitances are usually not a problem in most electronic circuits, but since our capacitances in the oscillator circuit are small (picofarads), parasitic capacitances become more significant.

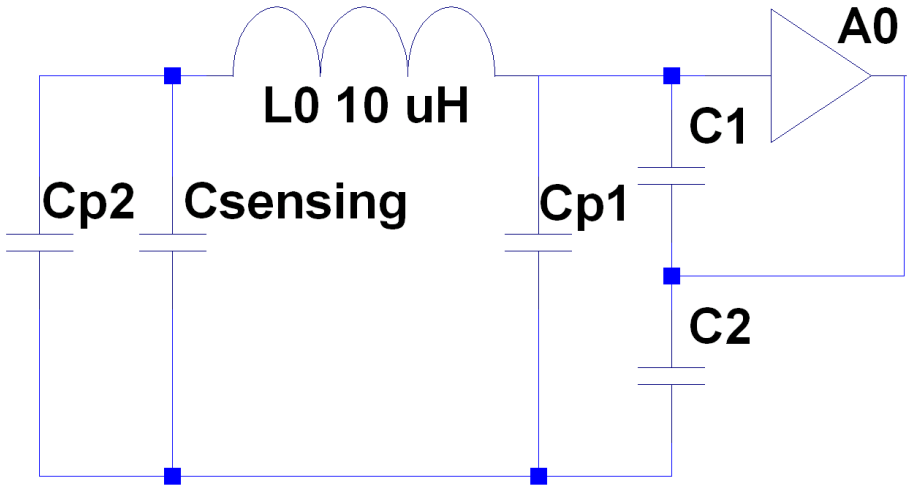


Figure 7. Oscillator schematic showing parasitic capacitances. A_0 represents the active component i.e. the bipolar transistor.

As mentioned before, a 10uH inductor was chosen for L0 (Figures 5a, 6, 7, 8, and 9) because this value is readily available in small sizes and it results in relatively small resonating capacitances at 16 MHz. For the Variable Frequency Oscillator to have a resonant frequency of 16 MHz, with a 10uH inductor, the total capacitance in the oscillator must equal 9.89pF (see equation 24). We selected a Clapp oscillator because we anticipated $C_{fb} >> C_{sensing}$, and Clapp oscillators transfer the change in capacitance of the smaller capacitor better than the other topologies. And with the frequency value we selected (16 MHz) and the inductor we selected (L0 = 10uH), the total capacitance in our circuit was9.89pF. To reach a total capacitance of 9.89 pF, when two capacitance are connected in series (C_{fb} and $C_{sensing}$ in Figure 5a), the smaller capacitance ($C_{sensing}$ in Figure 5a) must be higher than the desired total capacitance (9.89pF), hence $C_{sensing}$ must have a value of about 13pF and C_{fb} must

have a value around 40pF. C_{fb} was given a capacitance range of 35.68pF to 37.82pF (the trimmer tri1 in Figure 8 is what causes the range), because the parasitic capacitance Cp1 (Figures 7 and 8) increases the total capacitance range on this side of L0 to about 38pF to 41pF. $C_{sensing}$ was given a capacitance range from 8.16pF to 12.09pF (the trimmer tri2 in Figure 7 is what causes the range), because the parasitic capacitance Cp2 (Figures 7 and 8) increases the total capacitance range on this side of L0 to about 10pF to 14pF. Together, C_{fb} and $C_{sensing}$ make the total capacitance in the oscillator have a range between 6.64pF and 9.16pF, which both parasitic capacitors increase the range to about 8.50pF to 11.00pF. This range of capacitance values allowed us to calibrate the oscillator to a value very close to 9.89pF depending on the local humidity and temperature.

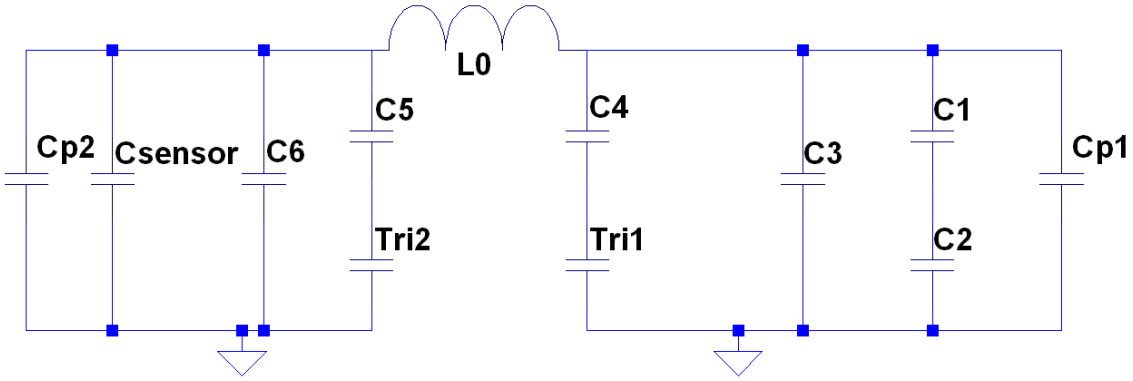


Figure 8. Schematic of complete Clapp oscillator used in the HumiSense as the sensing or variable frequency oscillator.

Recalling equations 23 and 32, the relation between F_0, L_0, and C_0 is:

$$F_0 = \frac{1}{2\pi\sqrt{L_0 C_0}} = F_{Osc} \text{ and } C_0 = \frac{1}{\left(\frac{1}{C_{sensing}} + \frac{1}{C_{fb}}\right)} = \frac{C_{fb} * C_{sensing}}{C_{sensing} + C_{fb}}$$

where L_0 is 10uH and C_0 is 9.89pF.

Then from the schematic on the oscillator on Figure 7, we generate the following relation for $C_{sensing}$ and C_{fb}:

$$C_{sensing} = \frac{1}{\left[\frac{1}{C_5} + \frac{1}{Tri_2}\right]} + C_6 + C_{sensor} + C_{P2} \tag{34}$$

and

$$C_{fb} = \frac{1}{\left[\frac{1}{C_1} + \frac{1}{C_2}\right]} + C_3 + \frac{1}{\left[\frac{1}{C_4} + \frac{1}{Tri_1}\right]} + C_{P1} \tag{35}$$

where $C_{sensing}$ is about 13pF and C_{fb} is about 40pF.

To achieve the capacitance range of 35.68pF to 37.82pF for C_{fb}, both feedback capacitors were selected to be 33pF (remembering that for the oscillator to work properly these two capacitors needed to each have a capacitance value of 27pF to 47pF), the fixed capacitor chosen had a capacitance value of 15pF, and the digital trimmer Tri1 (with an adjustable range from 6.41pF to 13.33pF) was connected in series with a 12pF fixed capacitor, so that the trimmer capacitance range would be 4.18pF to 6.32pF. It was desirable to decrease the trimmer capacitance range span significantly, to allow for each increase in value to be finer since this trimmer was our "fine" tuning trimmer for when calibrating the oscillator.

To achieve the capacitance range of 8.16pF to 12.09pF for $C_{sensing}$, the calculated sensor capacitance of 3.43pF was used (Note: this value was not confirmed independently), the fixed capacitor C6 was given a capacitance value of 1pF, and the digital trimmer (Tri2 with an adjustable range from 6.41pF to 13.33pF) was connected in series with a 18pF fixed capacitor C5, so that the trimmer capacitance range would be of 4.73pF to 7.66pF. The trimmer capacitance range of Tri2 was also decreased significantly, but not as much as with the Tri1's capacitance range, because this Tri2 was our "coarse" tuning trimmer. Figure 8 displays the names and capacitor values used on the final board.

The reason for having a "fine" and a "coarse" trimmer is to tune the total capacitance to as close to 9.89 pF as possible under all different conditions. The entire capacitance range of the fine tuning trimmer, Tri1, provides for a change of frequency of about 77.35 kHz , which is about equal to the frequency change caused by one linear step change of the coarse trimmer, Tri2 (73.95 kHz).

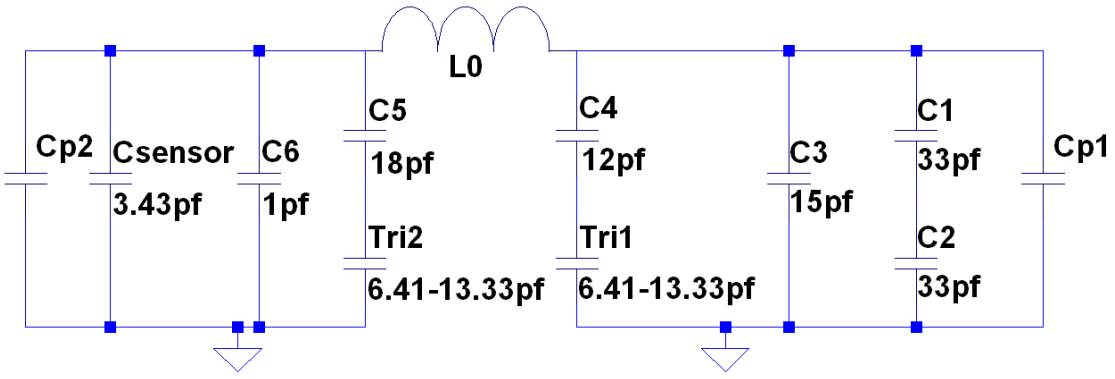


Figure 9. Variable Frequency resonator with sensing capacitor Csensor, the digital trimmers Tri1 and Tr2, the fixed capacitor C3, C4, C5, and C6, and the two positive feedback capacitors C1 and C2.

3.5.7.1 Digital Trimmers

Both digital trimmers were added to the circuit to tune the circuit as close to 16 MHz as possible regardless of the humidity, temperature, and pressure conditions at the moment that can affect the capacitance of the sensor capacitor. The trimmer that is part of the C_{sense} network (Tri_2) due to its location causes coarser shifts in the resonance frequency. Meanwhile, the trimmer that is part of the C_{fb} network (Tri_1) causes finer shifts in the resonance frequency. Together they allow the user to modify the resonance frequency of the variable frequency oscillator to within range of 16 MHz.

Both trimmers are Miniature Electronically Trimmable Capacitors (Maxim MAX1474). The MAX1474, when connected as one trimmable capacitor, had a range of capacitance that goes from 6.41pF at the lowest setting to 13.33pF at the highest setting. There were 32 possible settings (i.e. indexed from 0 to 31) with a linear step size of 0.216pF per unit increase in index. An adequate delay was used in the software code to account for any overshoot or ringing during transitioning of the trimmer's index value.

3.5.8 Sensor sensitivity computations

For purposes of easing computations, the sensor sensitivity analysis can be done on a model of the resonator where there is one single capacitor on the active site of the inductor and two capacitors (the sensor and a fixed capacitor) on the sensor side. Figure 8 is transformed to Figure 10:

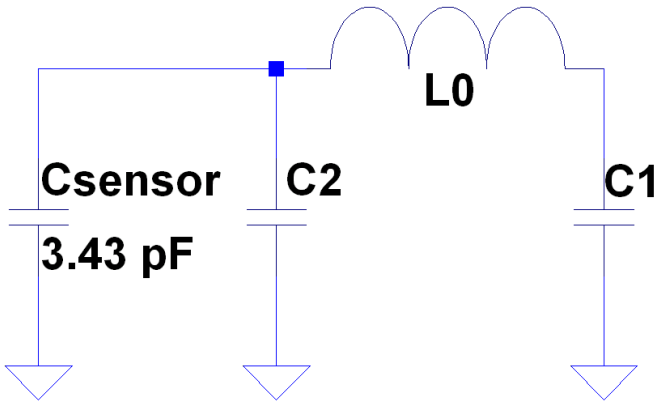


Figure 10. Circuit representation of the model of the variable frequency resonator where C_2 is the net capacitance parallel to C_{SENSOR} (C_S) and C_1 is the net capacitance on the active site of the inductor.

To further ease the sensitivity computations, we can combine C_{SENSOR} (C_S) and C_2 to C_2' (Figure 11), and then the total capacitance (C_t) will be equal to the series combination of C_1 and C_2':

$$C_t = \frac{C_1 * C_2'}{C_1 + C_2'} \tag{36}$$

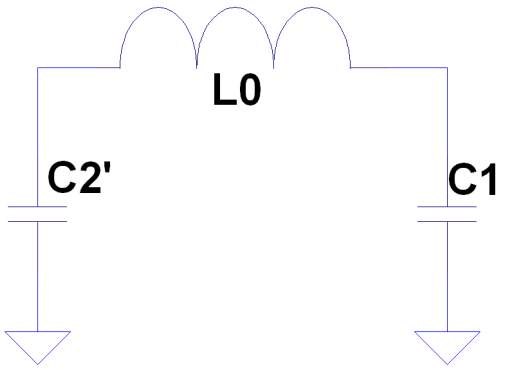


Figure 11. Circuit representation of the model of the variable frequency resonator where C_2 is the net capacitance parallel to C_{SENSOR} and C_1 is the net capacitance on the active site of the inductor.

In any LC oscillator (including the Clapp oscillator) with a total capacitance of C_T and a total inductance of L_0, the relation between the frequency change ratio and the capacitance change ratio, as mentioned on pages 25 and 26, can be derived as follows:

$$F = \frac{1}{2\pi\sqrt{L_0 * C_T}}$$

$$= \frac{C_T^{-1/2}}{2\pi\sqrt{L_0}}$$

Then,

$$\frac{dF}{dC_T} = -\frac{1}{2}\frac{C_T^{-\frac{3}{2}}}{2\pi\sqrt{L_0}}$$

$$= -\frac{1}{4\pi\sqrt{L_0 C_T^3}}$$

And,

$$\frac{dF}{F} = -\frac{dC_T}{4\pi\sqrt{L_0 C_T^3}} * \frac{2\pi\sqrt{L_0 * C_T}}{1}$$

$$= -\frac{1}{2} * \frac{dC_T}{C_T} \tag{37}$$

And the fractional changes in total capacitance (dC_T/C_T) can be derived from:

$$\frac{dC_T}{dC_2{}'} = \frac{d\left[\frac{C_1 * C_2'}{C_1 + C_2'}\right]}{dC_2'}$$

$$= \frac{C_1 * (C_1 + C_2') - (C_1 * C_2')}{(C_1 + C_2')^2}$$

$$= \frac{C_1^2}{(C_1 + C_2')^2}$$

where

$$\frac{dC_T}{C_T} = \frac{\frac{dC_2' * C_1^2}{(C_1 + C_2')^2}}{C_T}$$

$$= \frac{\frac{dC_2' * C_1^2}{(C_1 + C_2')^2}}{\frac{C_1 * C_2'}{C_1 + C_2'}}$$

$$= \frac{dC_2'}{C_2'} * \frac{C_1}{(C_1 + C_2')} \tag{38}$$

This means that a change in total capacitance is equal to the change in the total capacitance on the sensor side of the inductor times the ratio $C_1/(C_1+C_2')$, which is always less than 1. The ratio will approach 1 if C_2' is minimized or C_1 is maximized always being constrained by the resonant frequency and the inductor.

Now, a change in the sensor's capacitance can be seen in the comprehensive capacitance C_2' as follows:

$$\frac{dC_2'}{dC_S} = \frac{d(C_2 + C_S)}{dC_S}$$

$$= \frac{d(C_S)}{dC_S} = 1 \tag{39}$$

where C_2 has a fixed (or constant) value.

Then,

$$\frac{dC_2'}{C_2'} = \frac{dC_S}{C_2'}$$

$$= \frac{dC_S}{(C_2 + C_S)}$$

$$= \frac{dC_S}{C_S} * \frac{C_S}{(C_2 + C_S)} \tag{40}$$

Finally, if we recall equation (38), we can calculate what a change in the sensor capacitance will do to the total capacitance:

$$\frac{dC_T}{C_T} = \frac{dC_2'}{C_2'} * \frac{C_1}{(C_1 + C_2')}$$

If we now replace in equation (40), we get:

$$\frac{dC_T}{C_T} = \frac{dC_S}{C_S} * \frac{C_S}{(C_2 + C_S)} * \frac{C_1}{(C_1 + C_2 + C_S)} \tag{41}$$

This means that every change in sensor capacitance will be translated to the total capacitance of the circuit as the change in sensor capacitance times the ratio $\frac{C_S}{(C_2+C_S)} * \frac{C_1}{(C_1+C_2+C_S)}$ and this ratio can be controlled to a point so that the change in total capacitance due to the change in sensing capacitance can be maximized.

3.5.9 HumiSense's Software Operation and Counting Algorithm

The differential signal from the mixer was first passed through a low pass filter and then connected to the analog comparator inputs. The low pass filter eliminated the

addition signal coming from the mixer (i.e. the high frequency signal generated from the addition of the frequencies of the two input signals). The Analog Comparator was enabled as default. The *Analog Comparator Input Capture* bit and the *Timer/Counter1 Input Capture interrupt* bit were enabled, allowing the Input Capture function in Timer/Counter1 to be triggered by the analog comparator's output signal. The *Analog Comparator Interrupt Mode* chosen was the *Comparator Interrupt on Rising Output Edge* mode. This meant that whenever the signal connected to the positive input was higher than the signal on the negative input during the rising edge, the input capture register ICR1 acquired the *Timer/Counter 1* 's value (number of counts) at that moment.

The minimum period possible was limited by the time it takes to run all the instructions in the interrupt service routine (ISR). At its final revision, the ISR took 218 cpu clock cycles, in other words, the ISR is taking about (218*62.5nsecs=) 13.625usecs. The reciprocal of the ISR execution time equals the maximum input frequency (1/13.625 usecs = 73.395 kHz) that HumiSense can correctly log. The maximum number of counts is theoretically limited by the fact that the *Timer/Counter1* is a 16 bit counter and the largest decimal value a 16 bit integer can hold is 65,535. Thus, theoretically the maximum number of counts possible is 65,535.

There were two options in which to input the mixed frequency signal into the counter in the microcontroller unit. The signal could either be connected directly to the counter

or the signal could be connected to the counter as a differential signal. Since the output of the mixer is differential, the output signal was connected as a differential signal into the counter. There were two benefits of connecting our output as a differential signal. The first benefit was that there was a differential comparator available in the microcontroller, thus it removed the need for additional circuitry. And the second benefit was that differential sensing greatly reduces spurious responses due to DC voltage level shifts resulting from environmental changes.

The counter used had three inputs: control (CTRL), clock (CLK), and reset (RES). When the CTRL signal was “On”, the counter started counting clock pulses till the CTRL signal was turned “Off”. This counting mechanism can be used to “read” or “count” a signal by measuring period, frequency, or a mixture of both period and frequency.

3.5.9.1 Period counting

The signal’s period was determined by counting the number of clock periods between two consecutive clock edges of the same slope. The counting was done in a hardware timer clocked by the CPU clock frequency of 16 MHz. At each defined edge, the counter value was captured in hardware. Then, the count difference between successive edges indicated the number of clock intervals in the signal period.

To be able to use the counter to obtain the period of the mixer output signal, the output signal's period must be at least double the period of the clock pulse. In other words, the frequency of the output signal may be up to a maximum of half the clock frequency (Nyquist sampling rate theorem). On the other hand, the minimum frequency of the signal will be equal to the clock frequency divided by the maximum amount of counts possible. Every counter has a maximum number of counts equal to:

$$M_{max} = 2^n - 1 \tag{42}$$

where *n* is the number of bits.

Thus, the 16 bit counter found in our microcontroller could count up to:

$$M_{max} = 2^n - 1$$
$$= 2^{16} - 1$$
$$= 65535\ counts$$

And when using a 16 bit counter with a 16 MHz clock, the minimum measurable frequency is:

$$F_{min} = \frac{F_{clk}}{M_{max}}$$
$$= \frac{16MHz}{65535}$$
$$= 244.144Hz \tag{43}$$

Even though theoretically the counter can count changes of as low as 244 Hz, due to the signal's drift, we did not allow the heterodyne signal to be smaller than about 15kHz. 244.144Hz is not the resolution of this algorithm. The resolution is equal to the difference in frequency between the frequency generated by the current count (M) and the frequency generated by one count below it (M-1). So, when M is M_{max}, M-1 is M_{max-1}, and the difference in frequencies is the following:

For M_{MAX}:

$$F = \frac{F_{clk}}{M_{max}}$$
$$= \frac{16MHz}{65535}$$
$$= 244.144Hz$$

For M_{MAX-1}:

$$F = \frac{F_{clk}}{M_{max-1}}$$
$$= \frac{16MHz}{65534}$$
$$= 244.148Hz$$

$$Resolution = 244.148Hz - 244.144Hz = 0.004\ Hz \quad (44)$$

The resolution was 0.004 Hertz which made this option very accurate and also attractive. But, this was the maximum resolution of this method. The minimum resolution would be found when M was at its minimum, in other words, when it equaled two counts:

For M_{MIN}:

$$F_{max} = \frac{F_{clk}}{M_{min}}$$

$$= \frac{16.00\ MHz}{2}$$

$$= 8.00\ MHz$$

For M_{MIN+1}:

$$F = \frac{F_{clk}}{M_{min+1}}$$

$$= \frac{16.00\ MHz}{3}$$

$$= 5{,}333{,}333.33Hz$$

$$Resolution = 8.000MhHz - 5.333MHz = 2.667MHz$$

This means that the resolution decreased as the mixer output signal increased since the larger the mixed frequency signal was, the lower the number of counts the counter logged. So, choosing a range for the mixed signal frequency was necessary.

3.5.9.2 Frequency counting

Another method to measure the output signal with a counter would be to obtain the signal's frequency directly. Frequency counting was accomplished by counting the number of same slope edges (or events) in a specific time interval. The resolution of this method was not nearly as good as the resolution of the period method; this is because the resolution is a fixed resolution and it depends on the gate time which is the inverse of the frequency. We selected a gate time equal to the period of 40 Hz, which was 0.025 secs, because 40 Hz was our desirable sample rate. Since the counter had 16 bits of resolution, the maximum measurable frequency was:

$$\begin{aligned} F_{max} &= \frac{M_{max}}{T_{gate}} \\ &= \frac{65{,}535}{T_{gate}} \\ &= \frac{65{,}535}{0.025 secs} \\ &= 2{,}621{,}400\ Hz \end{aligned} \tag{45}$$

If we decreased the counts by one, the mixed frequency would be:

$$\begin{aligned} F_{max} &= \frac{M_{max-1}}{T_{gate}} \\ &= \frac{65{,}534}{T_{gate}} \end{aligned}$$

$$= \frac{65{,}534}{0.025secs}$$

$$= 2{,}621{,}360\ Hz$$

Giving a resolution of:

$$Resolution = |2{,}621{,}400\ Hz - 2{,}621{,}360\ Hz| = 40Hz$$

The minimum measurable frequency was:

$$F_{min} = \frac{M_{min}}{T_{gate}}$$

$$= \frac{1}{T_{gate}}$$

$$= \frac{1}{0.025secs}$$

$$= 40\ Hz$$

If we increased the counts by one, the mixed frequency would be:

$$F_{max} = \frac{M_{min+1}}{T_{gate}}$$

$$= \frac{2}{T_{gate}}$$

$$= \frac{2}{0.025secs}$$

$$= 80\ Hz$$

Giving a resolution of:

$$Resolution = |40\ Hz - 80\ Hz| = 40Hz$$

3.5.9.3 Period and frequency counting

For this project, the method selected was a mixture of period and frequency counting. This was chosen to improve the resolution of the frequency counting method at higher frequencies and to also be able to have a fixed baud rate for outputting the data. The fixed baud rate was selected at 40 Hz, assuming that it would be averaged down to a baud rate of 20 or 10 Hz since the current water vapor density sensors report their data at those baud rates. So, every 25ms a data point was reported by our board via serial to the main computer.

To implement the period and frequency counting algorithm, two counters were used in parallel. A 16 bit counter counted the number of main clock edges in between two mixed signal edges and added the number to a cumulative average. An 8 bit counter counted the number of mixed signal edges. The value of the cumulative average was compared to a threshold value we set at 331,776 (51,000 in hexadecimal format) every time there was a mixed frequency edge. If it were possible to count and transmit data simultaneously, the threshold chosen would have been 400,000, since 400,000 is equivalent to the total number of periods of the main clock (16 MHz) in 25ms. But

given that the microcontroller needed time between operations, we chose the threshold to be lower than 400,000, so that the microcontroller could operate and send a data point every 25ms. So, at the end the sensor would transmit two numbers: the cumulative average number, which would either be equal to the threshold (i.e. 331,776) or slightly higher, and the number of mixed frequency edges during the same amount of time in which the cumulative average was recorded.

The sensitivity of this counting method can be determined for specific mixed frequencies. For example, assume the mixed frequency (F_B) was 65 kHz. 65 kHz is a value on the high side for the mixed frequency range given that the range we went from 20 kHz to 73 kHz. Any signal lower than 20 kHz was too noisy and unstable to accurately log it. While 73 kHz corresponded to a period 13.7μsecs, which was the time it took to run the code. So, any signal with a period lower than 13.7μsecs would not be noticed. Now, since the reference frequency (F_r) was 16 MHz, the period of the clock (P_r) was the reciprocal of F_r. So, then if we were counting *N* reference periods, their total time length would be:

$$P_{total} = N * P_r = \frac{N}{F_r} \tag{46}$$

Since our algorithm was counting and adding the number of periods it takes to reach or exceed the threshold value of 331,776, we called this threshold value P_{TH}. It is important to remember that an input capture event happened every time the counter "saw" a direction-specific edge (we chose every rising edge) from the mixed

frequency oscillator (F_B). During each input capture, the number of reference frequency periods got added to the cumulative value and this cumulative value was compared against the threshold. So, to determine the number of capture events needed we can use the following relation:

$$N_{IC} * \frac{F_r}{F_B} = 331{,}776 \rightarrow N_{IC} = 331{,}776 * \frac{F_B}{F_r} \tag{47}$$

where N_{IC} is the number of Input Capture events.

But, since the microcontroller was set to only work with integers, the number of input capture events was equal to the integer value of the operation above plus one. So,

$$N_{IC} = INT\left(331{,}776 * \frac{F_B}{F_r}\right) + 1 \tag{48}$$

Then, if our mixed frequency was 65kHz, the number of input capture events needed to reach the threshold would be:

$$N_{IC} = INT\left(331{,}776 * \frac{F_B}{F_r}\right) + 1$$

$$= INT\left(331{,}776 * \frac{65{,}000Hz}{16{,}000{,}000Hz}\right) + 1$$

$$= 1{,}347 + 1 = 1{,}348$$

Now, to calculate the resolution, we must find out how much a change of one reference count will affect the mixed frequency. So, if we increased N_{IC} by 1, the mixed frequency would become:

$$N_{IC} = INT\left(331{,}776 * \frac{F_B}{F_r}\right) + 1 = 1{,}349$$

Then, the resolution would be equal to:

$$F_B = \frac{F_r * N_{IC}}{331{,}776} - 1$$

$$= \frac{16{,}000{,}000 * 1{,}349}{331{,}776} - 1$$

$$= 65{,}054.94\ Hz$$

A change of 1 count corresponds to a change of 55 Hz in the mixed differential frequency. This counting mechanism has a count maximum resolution of 48.23 Hz/count. This maximum resolution is not appropriate for our high frequency water vapor density sensor. Using the period counting algorithm and making sure the differential frequency was not higher than 6.9 kHz would have given us a resolution of 3Hz/count, which is higher than 1ppm resolution.

3.5.9.4 Is 55 Hz adequate?

A change in frequency of 55Hz would be product of a change in the sensing capacitor's dielectric constant, assuming all other parts remained constant. Equation (37) showed:

$$\frac{dF}{F} = -\frac{1}{2} * \frac{dC_T}{C_T}$$

where F is frequency and C_T is total resonator capacitance.

If the equation is rearranged, the change in total capacitance can be determined from the change in frequency:

$$\frac{dC_T}{C_T} = -2 * \frac{dF}{F} \tag{49}$$

This means that a change of 1 Hz in a 16 MHz oscillator would translate to a change in capacitance of:

$$\frac{dC_T}{C_T} = -2 * \frac{dF}{F}$$

$$= -2 * \frac{1\ Hz}{16\ MHz}$$

$$= -1.25E - 7$$

The value of the total capacitance of our resonator was determined from equation (24). The total capacitance was equal to 9.89 pF. This means that a change of 1 Hz in the 16 MHz resonator would be seen as a change in total capacitance equal to:

$$dC_T = -1.25E-7 * C_T$$
$$= -1.25E-7 * 9.89\,pf$$
$$= -1.23\,E-6\,pF\ per\ Hz\ of\ frequency\ change$$

A change in frequency of 55 Hz would imply a change in total capacitance of 67.65E-6 pF. The change needed in the sensor capacitance to give rise to a change of 67.65E-6 pF in the total capacitance can be calculated assuming all fixed capacitors have the values found on Figure 9 and the digital trimmers are set to their mid values. This means that the capacitors would have the following values: C_1 and C_2 = 33 pF, C_3 = 15 pF, C_4 = 12 pF, C_5 = 18 pF, C_6 = 1 pF, C_{p1} and C_{p2}= 2 pF, Tri_1 and Tri_2 = 9.72 pF, and C_{SENSOR}= 3.43 pF. On the active element side of the resonator, C_1 and C_2 were in series equating to 16.50 pF. C_4 was in series with Tri_1 totaling 5.37 pF. Then, C_{p1}, C_3, 16.50 pF, and 5.37 pF were in parallel, making the total capacitance of the active side of the resonator 38.87 pF.

On the sensor side of the resonator, C_5 and Tri_2 were in series equating to 6.31 pF. So then, C_{p2}, C_{SENSOR}, C_6, and 6.31 pF were in parallel, making the total capacitance on the sensor side equal to 12.74 pF. The total capacitance was made up by the series combination of the total capacitance of the active side and the total capacitance of the sensor side:

$$C_T = \frac{C_{active} * C_{sensing}}{C_{active} + C_{sensing}} \tag{50}$$

A change in total capacitance would then correspond to the following change in sensing capacitance:

$$\frac{dC_T}{dC_{sensing}} = \frac{d\left(\frac{C_{active} * C_{sensing}}{C_{active} + C_{sensing}}\right)}{dC_{sensing}} \tag{51}$$

After applying the quotient rule to the numerator of equation (51), the equation becomes:

$$\frac{dC_T}{dC_{sensor}} = \frac{d\left(\frac{C_{active} * C_{sensing}}{C_{active} + C_{sensing}}\right)}{dC_{sensor}}$$

$$= \frac{\frac{\left[d\left(C_{active} * C_{sensing}\right) * (C_{active} + C_{sensing})\right] - \left[d\left(C_{active} + C_{sensing}\right) * (C_{active} * C_{sensing})\right]}{(C_{active} + C_{sensing})^2}}{dC_{sensing}}$$

$$= \frac{\left[d\left(C_{active} * C_{sensing}\right) * (C_{active} + C_{sensing})\right] - \left[d\left(C_{active} + C_{sensing}\right) * (C_{active} * C_{sensing})\right]}{dC_{sensing} * (C_{active} + C_{sensing})^2}$$

Since C_{active} is a constant, the equation further reduces to:

$$\frac{dC_T}{dC_{sensor}} = \frac{\left[C_{active} * \left(C_{active} + C_{sensing}\right)\right] - \left(C_{active} * C_{sensing}\right)}{(C_{active} + C_{sensing})^2}$$

$$= \frac{C_{active}^2}{(C_{active} + C_{sensing})^2}$$

$$= \left(\frac{C_{active}}{C_{active} + C_{sensing}}\right)^2 \tag{52}$$

The change in sensor capacitance corresponding to a change seen in the total capacitance would then equate to:

$$dC_s = dC_T * \left(\frac{C_{active} + C_{sensing}}{C_{active}}\right)^2 \quad (53)$$

Given the values we selected for the trimmers, the change in sensor capacitance was:

$$dC_s = dC_T * \left(\frac{C_{active} + C_{sensing}}{C_{active}}\right)^2$$

$$= -1.23\ E-6 \frac{pF}{Hz} * \left(\frac{38.87\ pF + 12.74\ pF}{38.87\ pF}\right)^2$$

$$= -1.63E-6 \frac{pF}{Hz} = -1.63 \frac{aF}{Hz}$$

Given that C_{SENSOR} was approximately 3.43 pF, a change of 1.63 aF per Hertz of frequency change corresponded to a change:

$$\frac{dC_s}{C_s} = \frac{-1.63 \frac{aF}{Hz}}{3.43\ pF}$$

$$= -0.48E-6\ Hz^{-1}$$

From the previous section, the resolution of the counter was calculated to be 55 Hz per count. This means that the per count resolution of the sensor capacitance was equal to:

$$\frac{dC_s}{C_s} * counter\ resolution = -0.48E-6\ Hz^{-1} * 55\frac{Hz}{count}$$

$$= -26.40ppm\ count^{-1}$$

The current count resolution of 26.40ppm/count is not an adequate counting resolution to measure the changes in water vapor density of interest. If we recall section 3.3, changes of 1% relative humidity equate to changes of 1ppm in the sensor's dielectric constant. If changes of at least 1ppm in the sensor's capacitance cannot be measured, then changes of 1% relative humidity in the weather cannot be reported. And given that a 1% change in relative humidity corresponds to larger changes in absolute humidity when the temperature is high, the current HumiSense has even worse resolution in higher temperatures.

3.5.10 HumiSense logging application software

HumiSense Terminal was created by Dr. James Wagner for the purpose of logging the frequency events from HumiSense. The version 1.6 was the version used during the data collecting. The COM settings, the format for receiving data, and the Terminal Sample rate need to be configured for the software to work. Then, finally, the log file setting, New File or Appending to File, needs to be selected. A screenshot of the Graphical User Interface (GUI) of this software can be found in Appendix 5. If interested in the whole code, please contact Dr. James Wagner.

3.5.11 Thermistor Resistance Measurements for Board Temperature

It was desirable to know the board temperature, in order to offset the data that corresponded to changes in board temperature and not to external weather conditions. To capture the board temperature a surface-mount thermistor was soldered next to the variable frequency oscillator. The thermistor selected for measuring the board temperature was the Honeywell 173-103LAD-301. The thermistor had a resistance of 10Kohms (+/- 2% tolerance) at 25°C. The thermistor was connected to the Atmega168 microcontroller in a voltage divider configuration as follows:

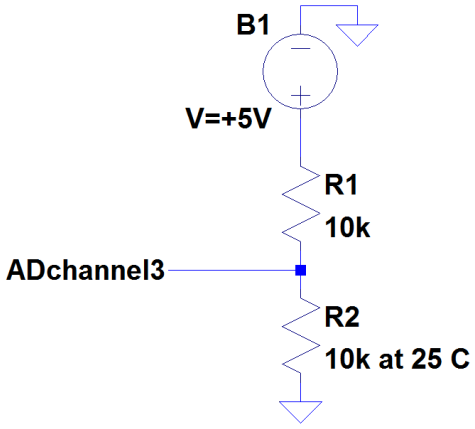


Figure 12. Thermistor circuit schematic. R1 is a 10kohm resistor and R2 is the thermistor. Both R1 and R2 make up the voltage divider circuit that gets input to the Analog to Digital converter channel 3 in the microcontroller.

The analog to digital converter (AD converter) will have a voltage at channel 3 between +4.2V and +1.2V corresponding to temperatures around -10°C and +50°C

respectively. These were the voltages values we expected to see, since our sensor would likely not be in temperatures outside that range. To find out what voltage the AD channel was experiencing at a specific time, the following equation must be used and rearranged to transform the digital AD value to an analog voltage value:

$$AD\ value = \frac{V_{IN} * 1024}{V_{REF}} \tag{54}$$

then,

$$V_{IN} = \frac{AD\ value * V_{REF}}{1024}$$

$$= \frac{AD\ value * +5V}{1024}$$

After finding V_{IN}, the resistance of the thermistor was determined by using the following voltage divider equation:

$$V_{IN} = \frac{R_{thermistor}}{R_{thermistor} + R_{REF}} * V_{REF}$$

then,

$$R_{thermistor} = \frac{R_{REF}}{V_{REF} - V_{IN}} * V_{IN}$$

$$= \frac{10k\Omega}{+5V - V_{IN}} * V_{IN}$$

Finally, to determine the board temperature, an equation was needed to approximate the thermistor. To determine this equation, the Resistance-Temperature table for the

thermistor was obtained (our thermistor curve was Honeywell's RT curve #16). Then, Excel was used to plot resistance against temperature. A least-squares fit algorithm built into Excel was used to fit the curve. A logarithmic trendline was selected and the equation corresponding to the trendline was obtained. The equation was:

$$T(°C) = -21.9 \ln(R_{thermistor}) + 227.64 \quad (55)$$

The R-squared value for the trendline fit was 0.997, so the resulting trendline matches the curve very closely and hence this equation was used to find the approximate value of the board temperature.

3.6 Steady state temperature, pressure, and humidity experiments

We conducted steady state experiments for temperature, pressure, and humidity conditions to test for the expected level of signal noise in HumiSense.

3.6.1 Setup for steady state temperature, pressure, and humidity of HumiSense

The HumiSense assembly was placed into a Precision Scientific Thelco gravity convection oven with a Campbell Scientific CR10X relative humidity and temperature probe. The HumiSense and probe were subjected to different temperatures in the range from 20°C to 50°C for periods of two hours in order to see if the sensor's frequency

signal remained constant, while the temperature was constant. This temperature signal was used as the expected "noise" signal of the HumiSense.

3.6.2 Setup for steady state temperature, pressure, and humidity without the capacitive sensor

For the second set of experiments, the HumiSense was placed in the same gravity convection oven with the CR10X relative humidity and temperature probe. However, the capacitive sensor was removed and two $8.2x10^{-12}$F NPO (negative positive zero) capacitors were added in parallel to replace the capacitive sensor. NPO capacitors were chosen because they have small temperature coefficients and are readily available. Two $8.2x10^{-12}$F capacitors in parallel appear to the rest of the circuit as a $4.1x10^{-12}$F capacitor with 5% tolerance. The HumiSense and probe were subjected to the same temperature settings as before, to see if the board components were the cause for most of the signal noise. This temperature signal was used as the expected "noise" signal of the circuit board components.

3.6.3 Setup for steady state temperature, pressure, and humidity without the capacitive sensor, while in a thermal insulation box

For the third set of experiments, the HumiSense was placed inside a 2cm thick thermally insulated box. Then, the assembly was placed into the same gravity convection oven with the CR10X relative humidity and temperature probe. The

capacitive sensor was replaced by the pair of $8.2x10^{-12}$F NPO capacitors. The HumiSense and probe were subjected to similar temperature settings to see if the thermally insulated box diminished the noise signal.

3.7 Field tests

The HumiSense was taken for field tests in October 2011, before we had done the steady state condition experiments. The HumiSense was placed 3 meters from the ground next to a Licor LI-7500A Open Path CO_2/H_2O Gas Analyzer. The LI-7500A recorded densities of carbon dioxide and water vapor (in units of mmol*m^{-3}) and also atmospheric pressure (in units of kiloPascals) at a rate of 10Hz. Virtual temperature readings (in units of °C) at10 Hz from two sonic anemometers, a Metek FHN-1 R.M. and a Young 81000 VRE, were first transformed to dry-bulb air temperature and then used with the water vapor density and atmospheric pressure data from the Licor to calculate the expected air dielectric constant using equation (18). The equation for the relation between virtual temperature and atmospheric temperature is the following:

$$T_V = (1 + 0.61 * q) * T \tag{56}$$

where T_V is virtual temperature in ºC, q is specific humidity in kg*kg^{-1}, and T is dry-bulb air temperature in ºC.

To obtain atmospheric temperature from the virtual temperature signal the following equation was used:

$$T = \frac{T_V}{1 + 0.61 * q} \tag{57}$$

where T_V is virtual temperature in ºC and q is the specific humidity in kg/kg. Specific humidity was first obtained from the following relation:

$$q = \frac{\rho_{watervapor}}{\rho_{air}} \tag{58}$$

where $\rho_{watervapor}$ is the density of water vapor in kg m^{-3} and ρ_{air} is the density of air in kg m^{-3}. The density of water vapor was obtained from the Licor and 1.2041 kg m^{-3} was used for density of air.

The water vapor pressure and the atmospheric pressure 10 Hz signals from the Licor LI-7500A were both used in equation (57) to obtain the 10 Hz atmospheric temperature. It is important to note that the 10 Hz atmospheric pressure signal is derived from a different sampling site in the instrument. The pressure is basically measured in the main box of the instrument instead of in the sensor. The water vapor density data was converted from mmol*m-3 to kg*m^{-3}, the atmospheric pressure data was converted from kPa to Pa, and the virtual temperature data was converted from °C to K. These converted values were then substituted into equation (18) to calculate the expected dielectric constant of air (results from one of the field test events are found in the results section). Then, the 10 Hz dielectric constant signal was inserted into equation 28 to calculate the expected coaxial capacitance for the 10 Hz signal. Next, this expected coaxial capacitance 10 Hz signal was inserted into equation 34, and the

expected $C_{sensing}$ capacitance signal was used to calculate the total capacitance (equation 23), assuming that all the other parts in equations 34 and 35 were constant (i.e. assuming no manufacturing variance).

The high frequency variation in the total calculated capacitance obtained from the 10 Hz water vapor, pressure, and temperature data was compared to the high frequency variation in the measured total capacitance from the HumiSense data log events by plotting both against time and by creating a correlation scatter plot. The same two plots were made for the first 10 seconds of the 10 Hz, 1Hz, and 0.1Hz capacitance signals. The correlation and R^2 coefficients were calculated for all the correlation scatter plots. All plots and calculations were created using the numerical computing environment, MATLAB.

MATLAB was also used to create a normalized power spectrum plot for each of the 10 Hz capacitance signals and a normalized power coespectrum plot of the correlation of both 10 Hz capacitances signals. Power spectrum plots are representations of a time-domain signal in the frequency domain. The power spectrum signal can be generated using a Fourier transform. Scalon et al. (2001) explain the use of power spectral density to characterize the turbulent transport of carbon dioxide and water and how Fourier transformation can be used to quantify the power spectral density. The normalized power spectrum plots of each 10 Hz capacitance signals will be compared. The signals were normalized by multiplying the by the frequency (i.e. the

corresponding value of frequency (x-axis value) for each specific capacitance value (y-axis value)) and dividing by the variance in the signal. Normalizing allows the comparison of multiple signals with different value ranges.

4 Results

Following, results from the steady state temperature experiments and the field tests will be shown.

4.1 Steady state temperature, pressure, and humidity

The steady temperature, pressure, and humidity experiments show the expected noise levels of HumiSense given that they were measured at constant temperature, pressure, and humidity.

4.1.1 Steady temperature, pressure, and humidity of Humisense

Figures 13 and 14 show the signal noise of the entire HumiSense system in units of Farads when the temperature, pressure, and humidity were kept constant. Figures 15 and 16 show the signal noise of the entire HumiSense system in units of Hertz when the temperature, pressure, and humidity were kept constant. Figure 13 and Figure 15 show the first 330 seconds and Figure 14 and Figure 16 show the last 300 seconds of the fixed environmental conditions experiment done on December 14th 2011. For Figures 13 through 16, the temperature in the oven was set to 22 °C and the vapor pressure was constant around 0.45kPa.

The oscillator was oscillating steadily for the entire time between 16.021 MHz and 16.022 MHz (i.e. the mixer output signal was steadily between 21 kHz and 22 kHz). The average capacitance noise of HumiSense is around +/-0.7 femtofarads. Figure 17 is the count distribution of the difference frequency noise in Figures 15 and 16. The 500 Hz noise spikes found in Figures 15 and 16 are not the large majority of the noise levels seen.

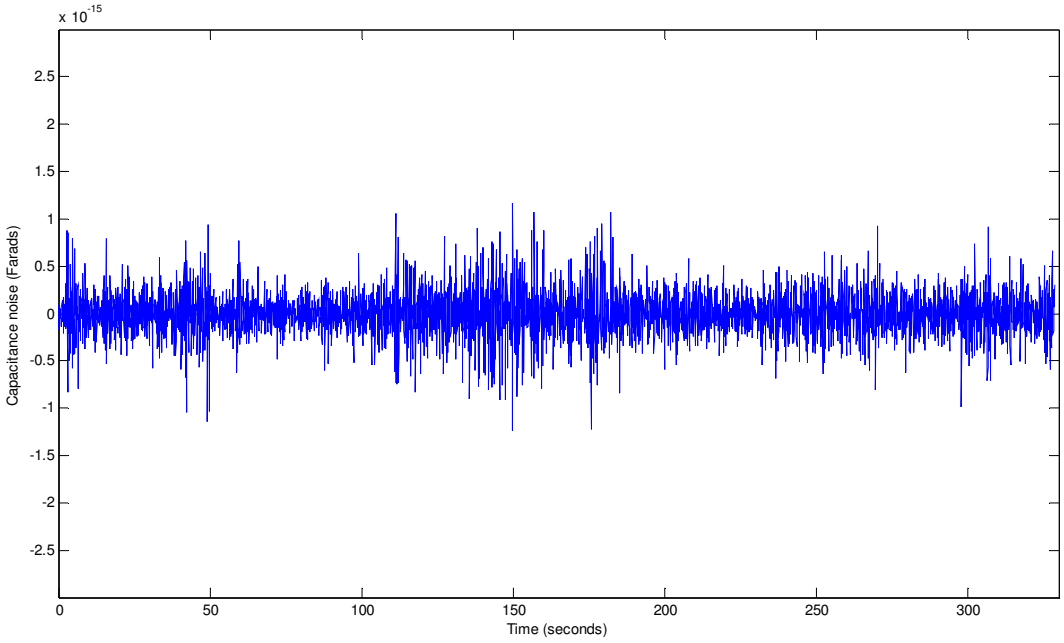


Figure 13. Signal noise of HumiSense in units of Farads. This figure shows the first 330 seconds of the fixed environmental conditions experiment done on December 14th 2011.

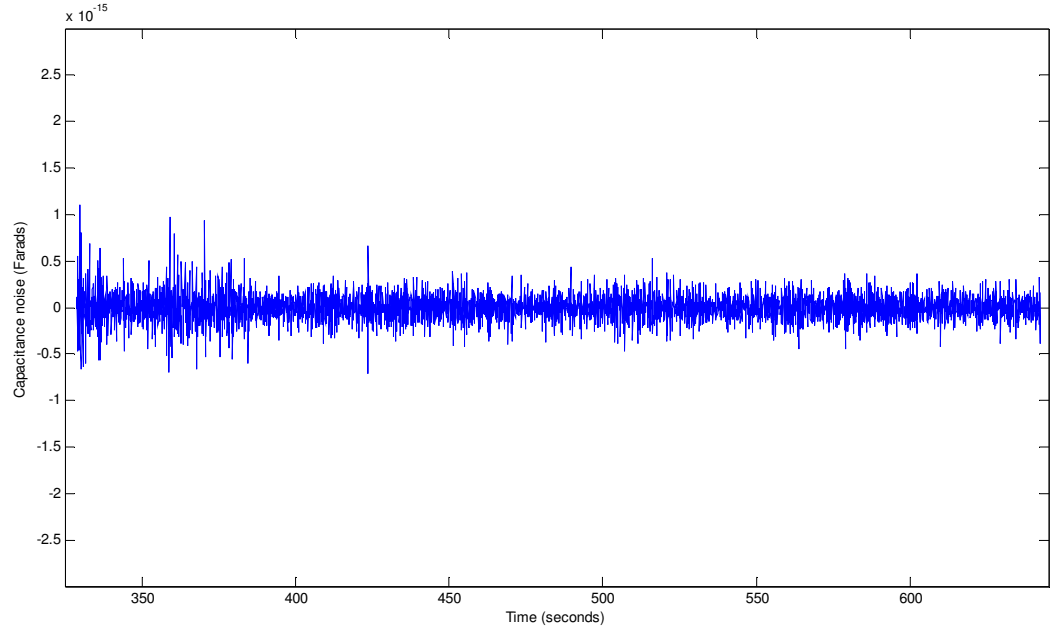


Figure 14. Signal noise of HumiSense (all parts, circuit board and coaxial capacitor were present) in units of Farads. This figure shows the last 300 seconds (from 330 to 630) of the fixed environmental conditions experiment done on December 14th 2011.

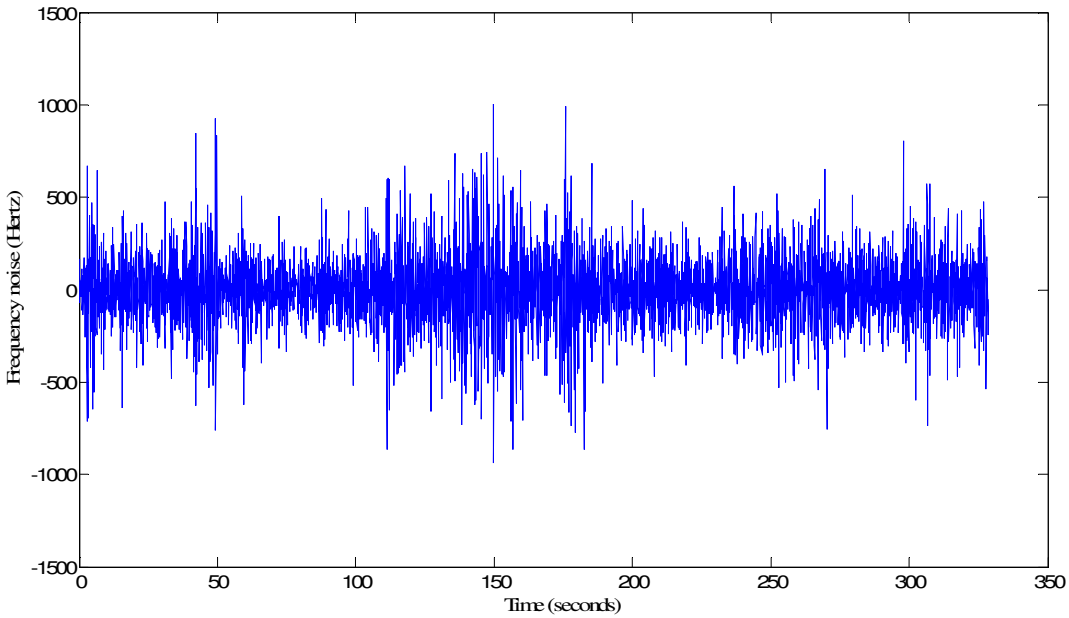


Figure 15. Signal noise of HumiSense in units of Hertz. This figure is the frequency version of Figure 13, so the same temperature and water vapor pressure conditions apply to this plot. The same 330 seconds of data from Figure 13 apply to this figure too.

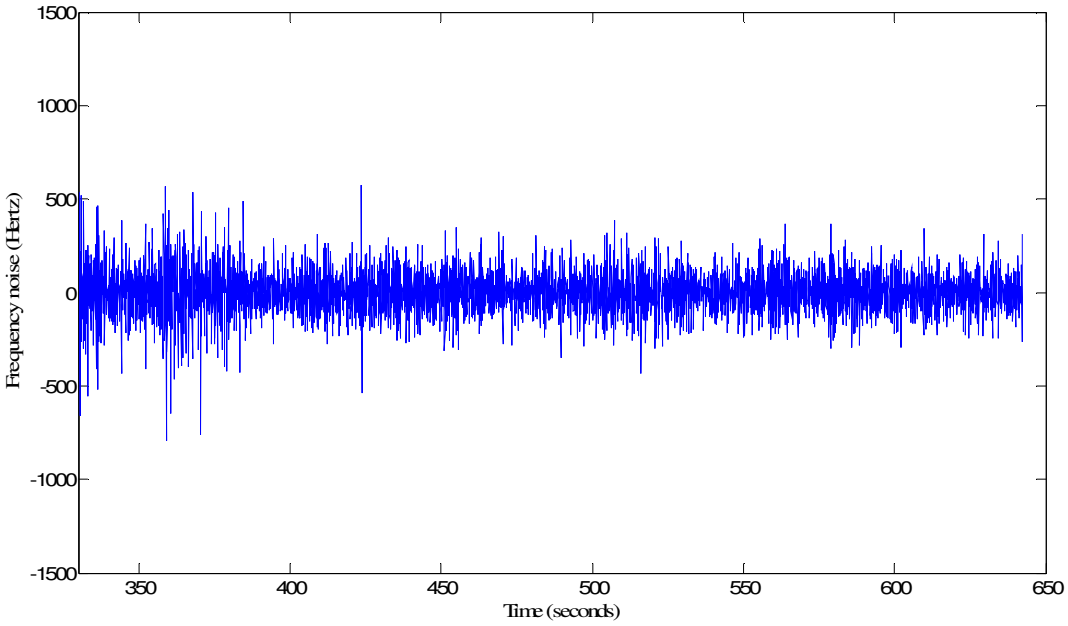


Figure 16. Signal noise of HumiSense in units of Hertz. This figure is the frequency version of Figure 14, thus the same temperature and water vapor pressure conditions apply to this plot. The same 300 seconds of data from Figure 14 apply to this figure too.

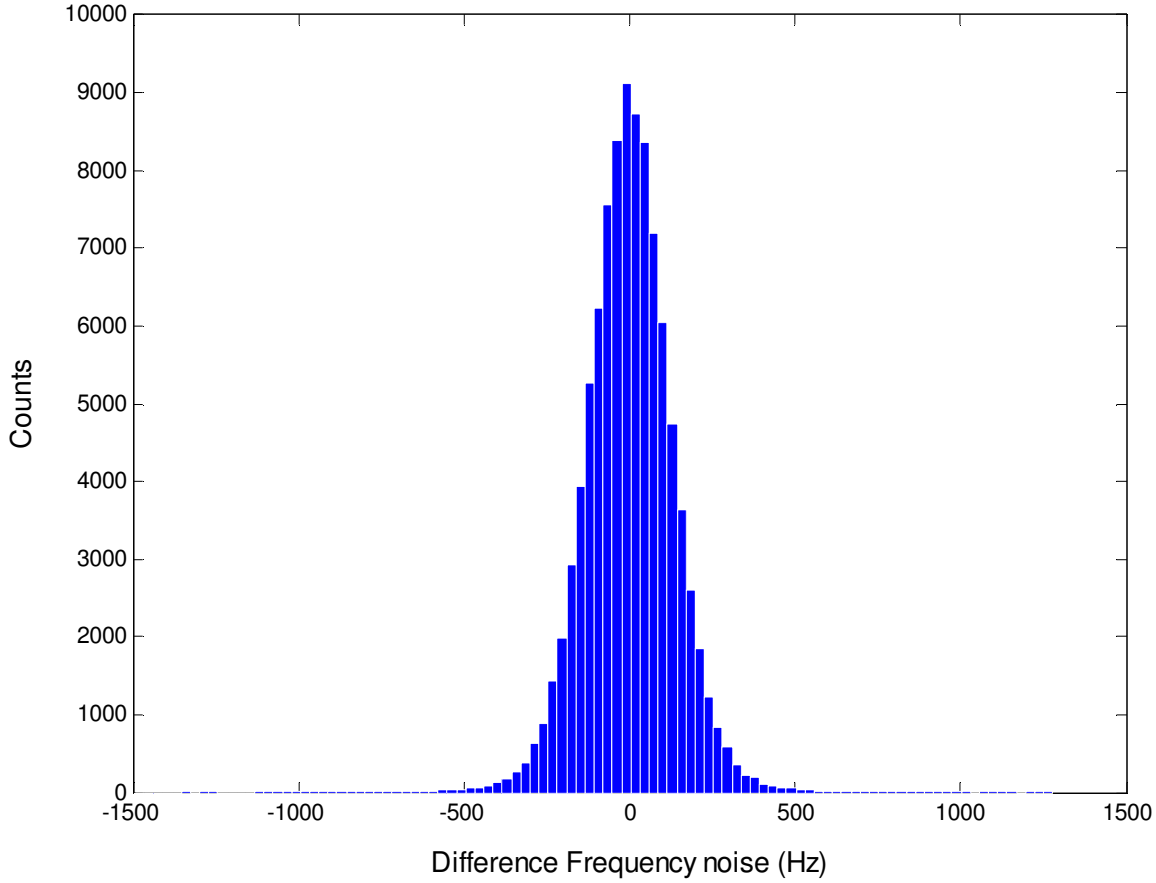


Figure 17. Count distribution of the difference frequency noise in Figures 15 and 16.

4.1.2 Steady state temperature, pressure, and humidity without the capacitive sensor

Figures 18 and 19 show the signal noise of HumiSense without the capacitive sensor in the circuit. Figure 18 is in units of Farads and Figure 19 is in unit of Hertz. Both show noise data when the temperature, pressure, and humidity were kept constant. The capacitive sensor was replaced by two 8.2×10^{-12}F NPO capacitors connected in parallel. Figure 18 and 19 show the last 630 seconds of the fixed environmental

conditions experiment done on January 10th 2012.This experiment was performed to see if most of the noise was attributable to the circuit board. The temperature in the oven was set to 22 °C and the vapor pressure was constant around 0.7kPa.

The oscillator was oscillating steadily for the entire time between 16.0205 MHz and 16.0225 MHz (i.e. the mixer output signal was steadily between 20.5 kHz and 22.5 kHz). The average capacitance noise of the circuit board is around +/-1.5 femtofarads making it larger than the average noise of the experiment with the coaxial sensor present. Figure 20 is the count distribution of the difference frequency noise in Figures 19. The 500 and 1000 Hz noise spikes seen in Figure 19 are visible, but again they are not the large majority of the noise levels seen. According to Figure 19, a single 2000 Hz noise spike is present. Nonetheless, this noise spike cannot be seen in Figure 20. When comparing Figure 20 to Figure 17, the variance in the central peak is lower in Figure 20, but the two “side-bands”, each centered at 1000 Hz, make the computed variance higher.

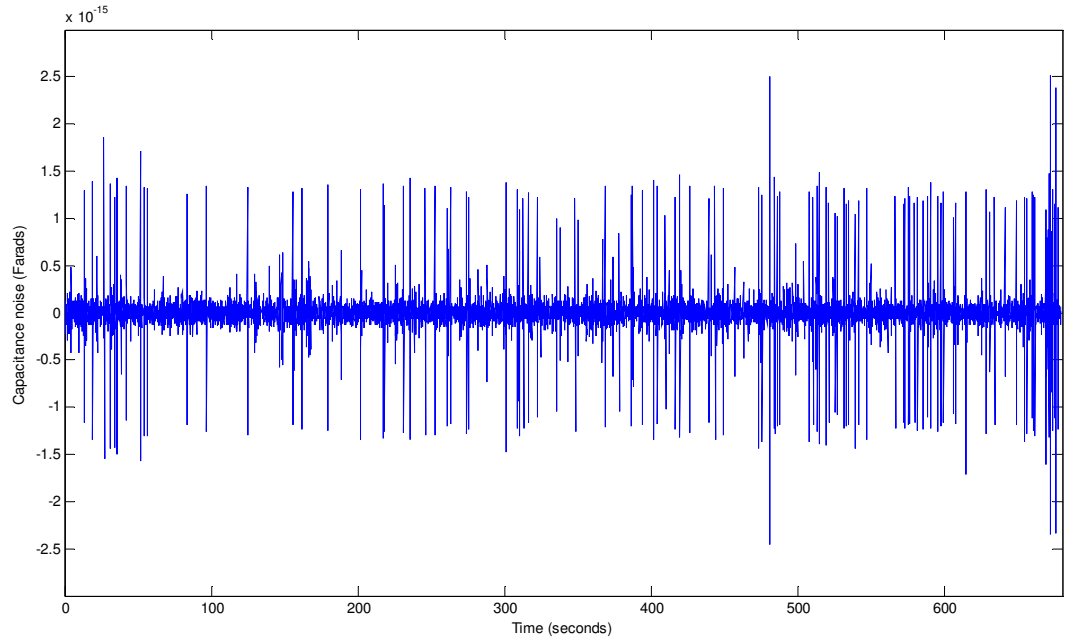


Figure 18. Signal noise of the circuit board in units of Farads. This figure shows the last 630 seconds of the fixed environmental conditions experiment done on January 10th 2012.

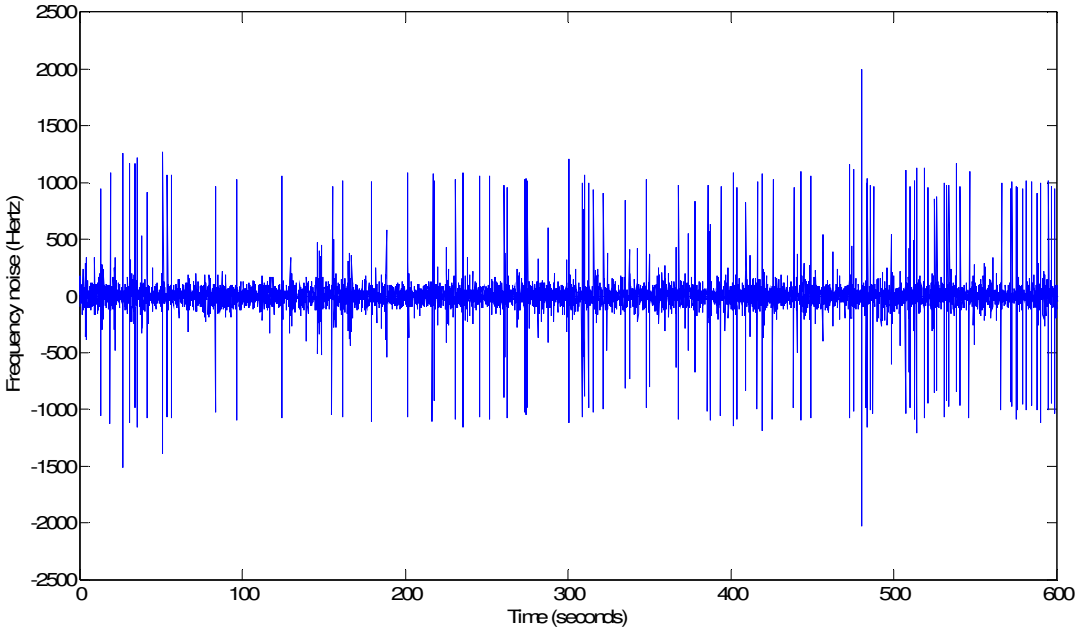


Figure 19. Signal noise of HumiSense in units of Hertz. This figure is the frequency version of Figure 18, so the same temperature and water vapor pressure conditions

apply to this plot. The same 630 seconds of data from Figure 18 apply to this figure too.

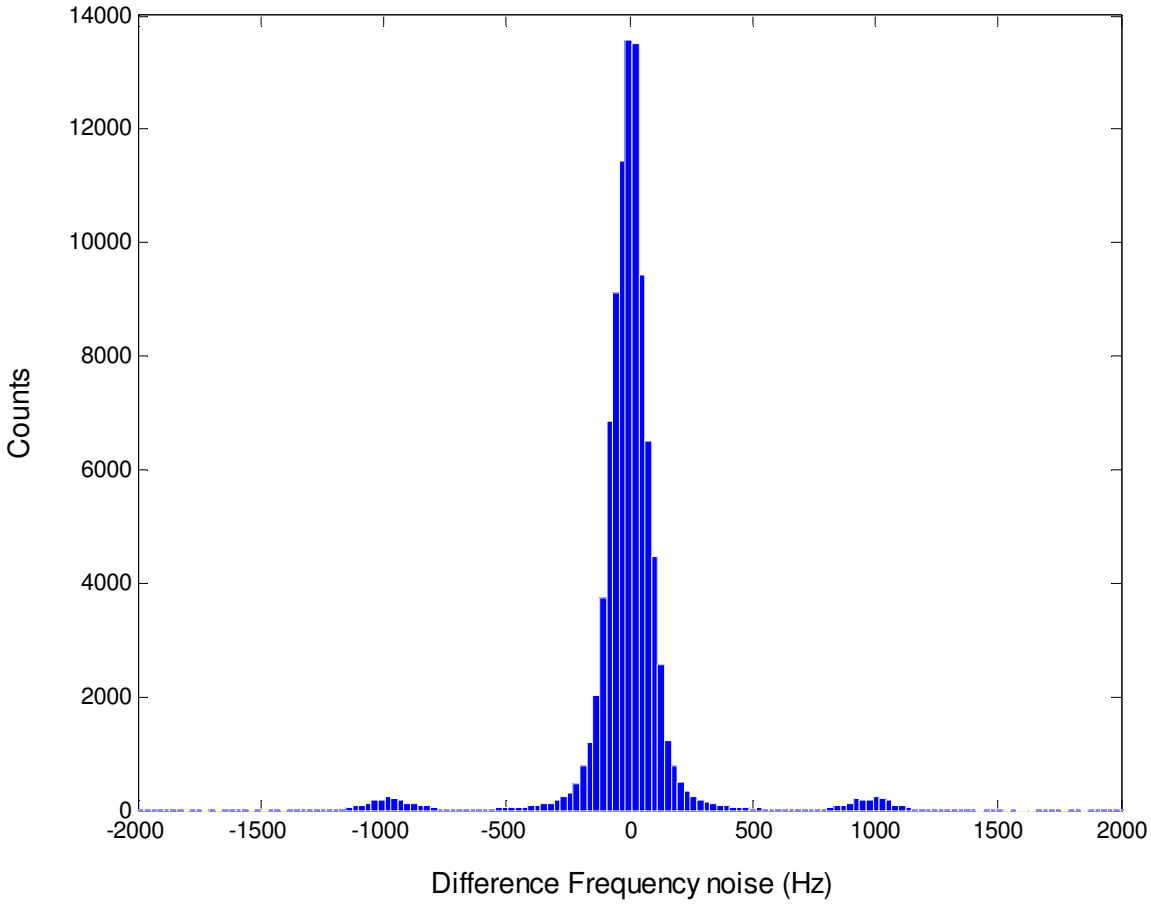


Figure 20. Count distribution of the difference frequency noise in Figure 19.

4.1.3 Steady state temperature, pressure, and humidity without the capacitive sensor while in a thermally insulated box

Figures 21 and 22 show the signal noise of HumiSense with the capacitive sensor removed and while inside a thermally insulated box. Figure 21 is in units of Farads and Figure 22 is in unit of Hertz. Both show noise data when the temperature, pressure, and humidity were kept constant. The capacitive sensor had been replaced by

two $8.2x10^{-12}$F NPO capacitors connected in parallel. Figures 21 and 22 show the last 630 seconds of the fixed environmental conditions experiment done on January 22nd 2012.This experiment was performed to see if the circuit board noise could be reduced by insulating the board. The temperature in the oven was set to 50 °C (higher than the other steady state condition experiments) and the vapor pressure was constant around 0.92kPa. The oscillator was oscillating steadily for the entire time between 16.020 MHz and 16.025 MHz (i.e. the mixer output signal was steadily between 30 kHz and 35 kHz).

The average capacitance noise of the circuit board is only about +/-3 femtofarads making it significantly higher than the average noise of the circuit board without insulation. The noise might have been higher because the temperature was higher than in the two previous experiments. Figure 23 is the count distribution of the difference frequency noise in Figures 22. The 1500 and 3000 Hz noise spikes seen in Figure 22 are present, but again they are not the large majority of the noise levels seen. And there are no more than five spikes around 5000 and 6000 Hz (see Figure 22), and they are not visible in Figure 23.

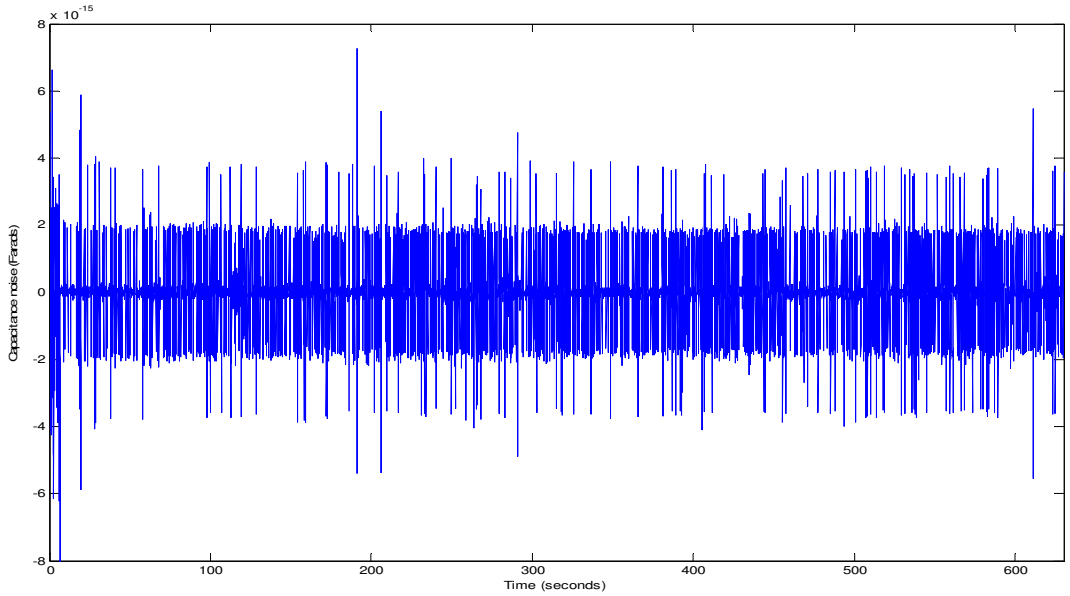


Figure 21. Signal noise of the circuit board inside a thermally insulated box in units of Farads. The coaxial sensor (i.e. the sensing capacitor) was not present for this experiment. This figure shows the last 630 seconds of the fixed environmental conditions experiment done on January 22nd 2012.

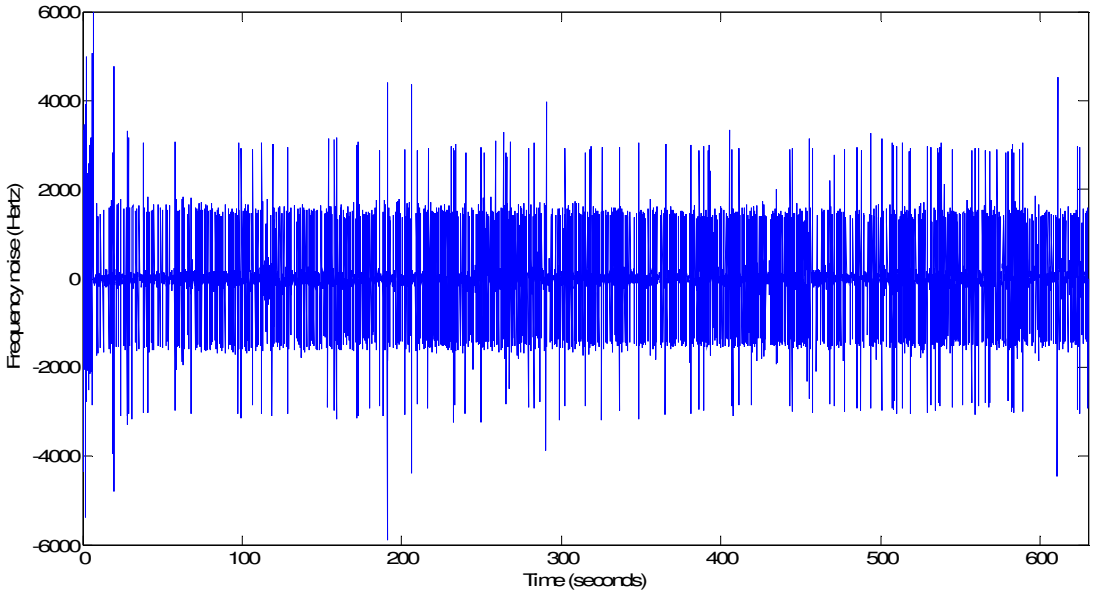


Figure 22. Signal noise of HumiSense in units of Hertz. This figure is the frequency version of Figure 21. The same 630 seconds of data from Figure 21 apply to this figure too.

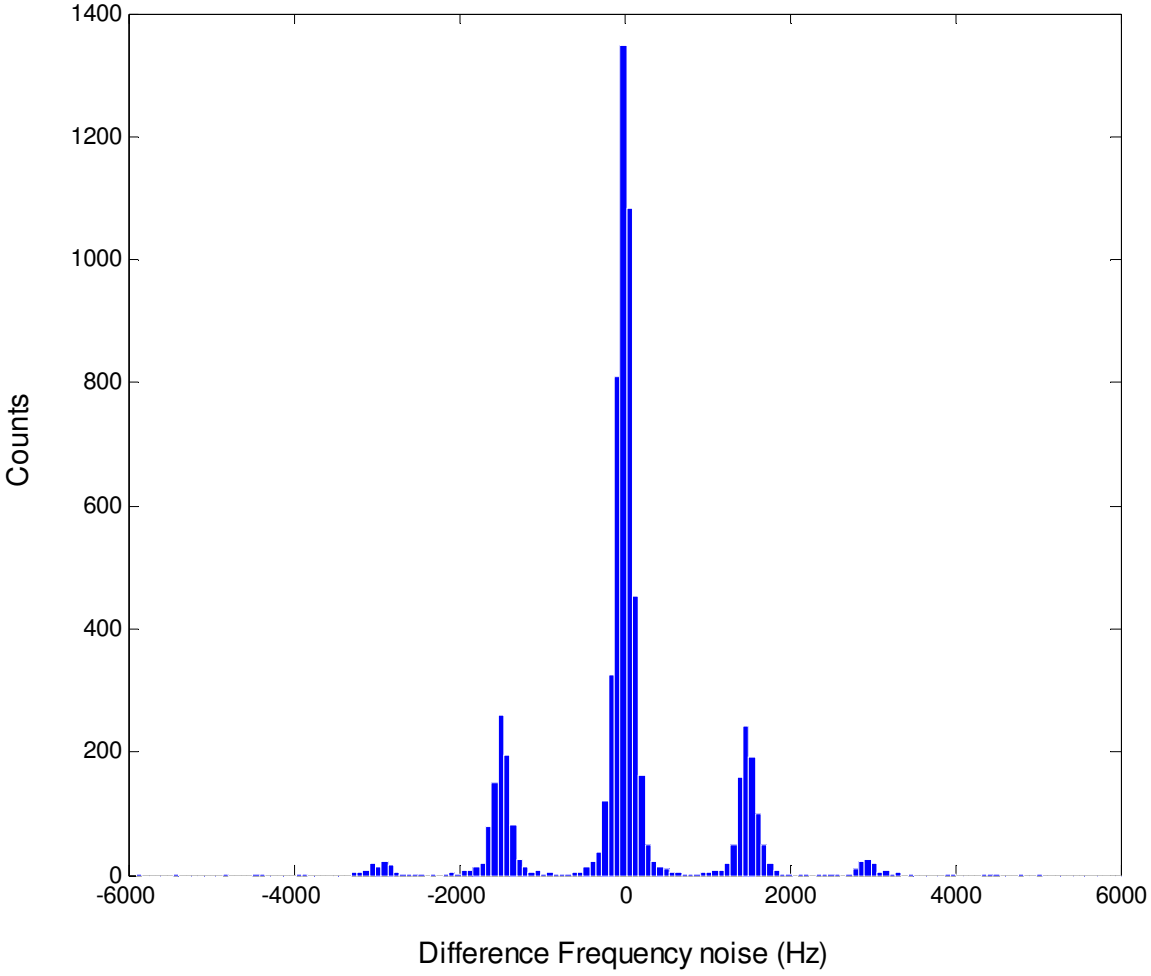


Figure 23. Count distribution of the difference frequency noise in Figure 22.

4.2 Field tests

Results from a field test deployment from October 6th 2011, are plotted on Figures 24, 26, 28, and 30. Figures 25, 27, 29, and 31 are the correlation plots of the signals shown on Figures 24, 26, 28, and 30 respectively. Figure 24 shows the measured 10 Hz capacitance signal in green and the 10 Hz expected total capacitance signal in blue.

The expected total capacitance was calculated from the Licor 10 Hz water vapor density data and 10 Hz atmospheric pressure data and also from the Metek sonic anemometer 10 Hz temperature data. The water vapor density, barometric pressure, and temperature 10Hz data was used to calculated the 10 Hz air dielectric constant signal. Then the 10 Hz dielectric constant signal was used in equation 33, assuming all other electronic parts had the values provided by their manufactureres, to calculate the expected C_{SENSE}. The calculated C_{SENSE} was used in equations 32 and 26 to calculate the expected total capacitance for HumiSense which is the signal displayed in green in Figure 24.

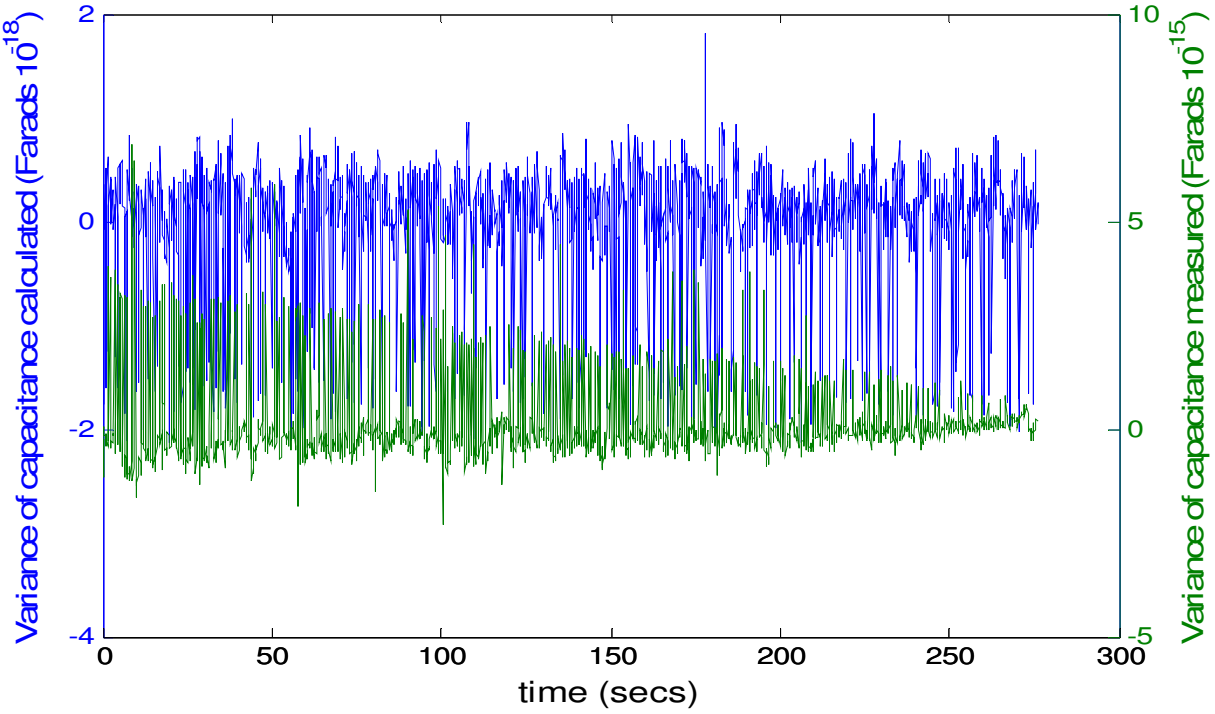


Figure 24. The 10 Hz measured capacitance signal of the total capacitance (in green) and the 10 Hz expected capacitance signal total capacitance (in blue).

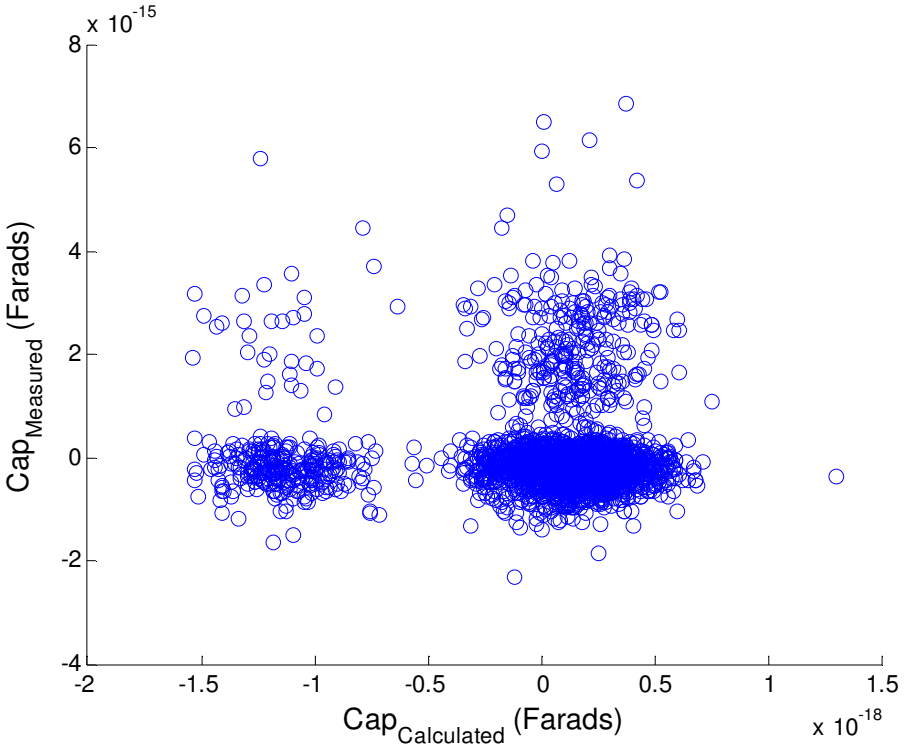


Figure 25. This figure shows the correlation of both capacitance signals shown in Figure 24.

Figure 25 is the correlation plot of the capacitance signals from Figure 24. There seems to be no evidence of correlation between the two signals. A perfect correlation scatter plot would show a near linear relationship between $CAP_{calculated}$ and $CAP_{measured}$ with slope either +1 or –1. The heavy vertical lines are an artifact of the data, in that, the scatter plot was generated from a time series with $CAP_{measured}$ having periods of high frequency variability. This artifact disappears as with a lower sampling rate.

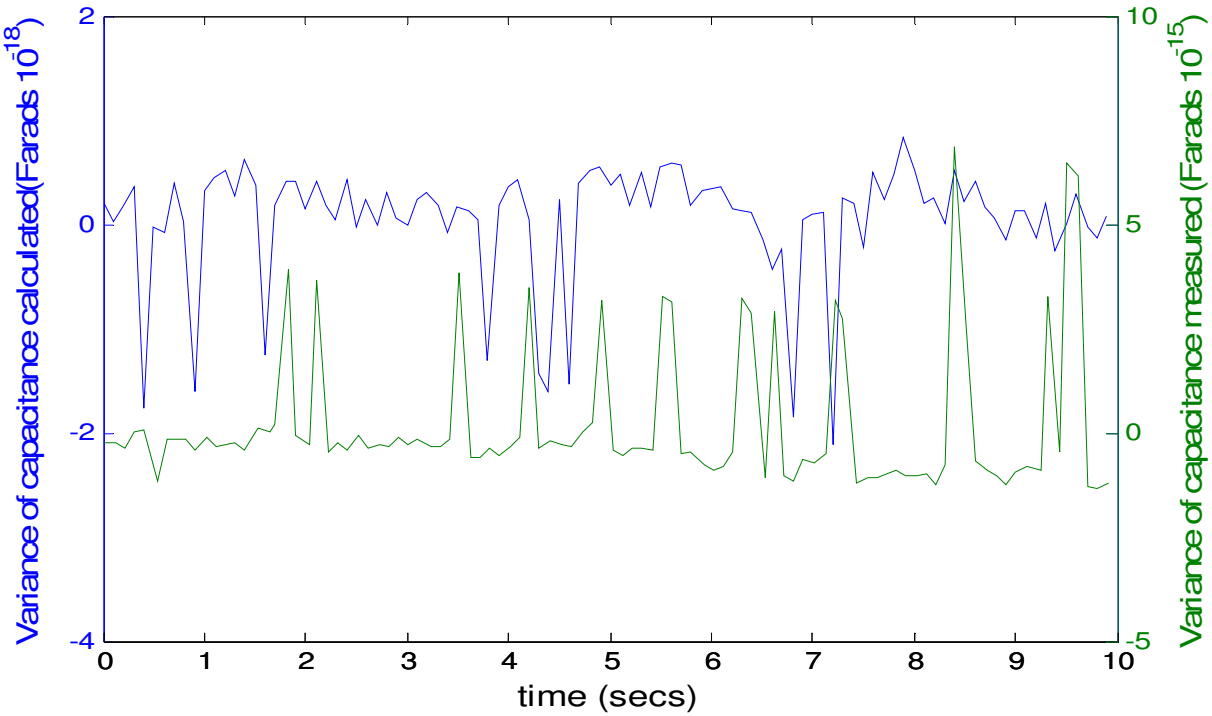


Figure 26. The first ten seconds of Figure 24.

Figure 26 is a closeup of the first ten second from Figure 24. Figure 27 is the correlation plot of the capacitance signals from Figure 26. Again, there is no evidence of correlation between the two signals.

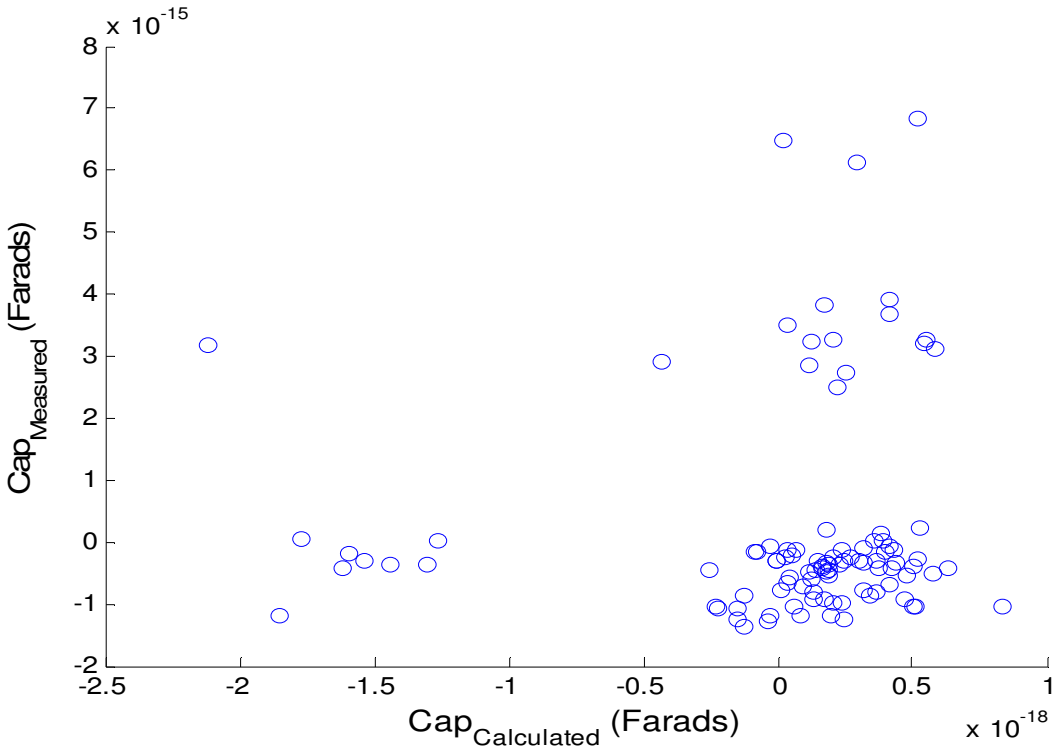


Figure 27. This plot shows the correlation of the first ten seconds of both signals on Figure 24.

Figure 28 shows the measured 1 Hz capacitance signal in green and the 1 Hz expected total capacitance signal in blue. Figure 30 shows the measured 0.1 Hz capacitance signal in green and the 0.1 Hz expected total capacitance signal in blue. The signals for Figure 28 were obtained by subjecting the data from Figure 24 to a running average with a 10 sample window. The signals for Figure 30 were obtained by increasing the window of the running average to a 100 sample window.

Figure 29 is the correlation plot of the capacitance signals from Figure 28. There seems to be no evidence of correlation between the two signals. Looking at Figure 29, it is harder to see any evidence of correlation than in Figure 24. In fact from MATLAB, autocorrelation between $CAP_{calculated}$ and $CAP_{measured}$ results in a value close to 0. Figure 31 is the correlation plot of the capacitance signals from Figure 30. For this Figure, it is even harder to see any evidence of correlation than in both Figures 25 and 29. The correlation coefficients and R^2 values were calculated for Figures 25, 27, 29, and 31. The correlation coefficient for Figure 25 was 0.0063, for Figure 27 it was 0.0615, for Figure 29 it was 0.0449, and for Figure 31 it was 0.1991. The R^2 values for Figures 25, 27, 29, and 31 were 0.000040, 0.003782, 0.002016, 0.039641. The R^2 values for Figures 25, 27, 29, and 31 are significantly lower than that expected from correlated signals.

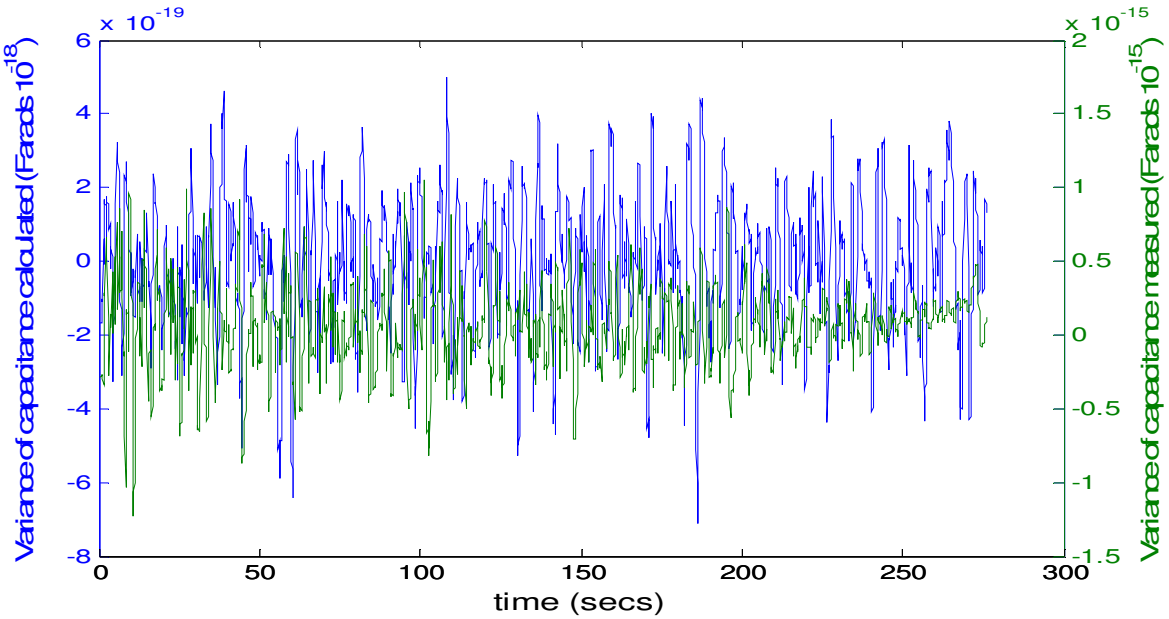


Figure 28. The 1 Hz capacitance signal of the total capacitance in HumiSense is shown in green. The 1Hz expected total capacitance of HumiSense is shown in blue.

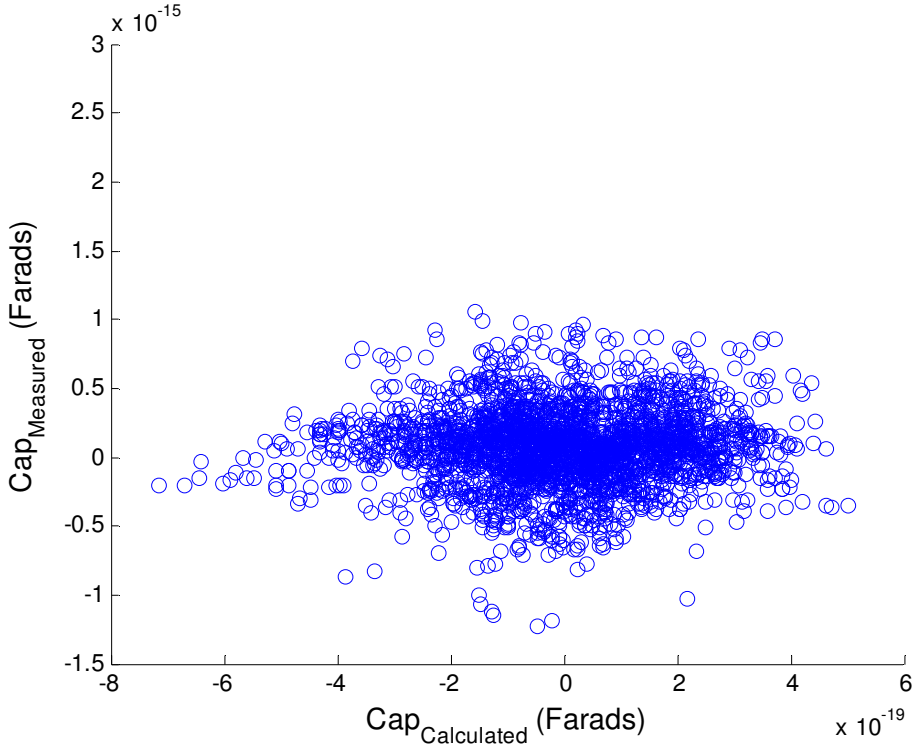


Figure 29. Correlation of \capacitance signals shown in Figure 28.

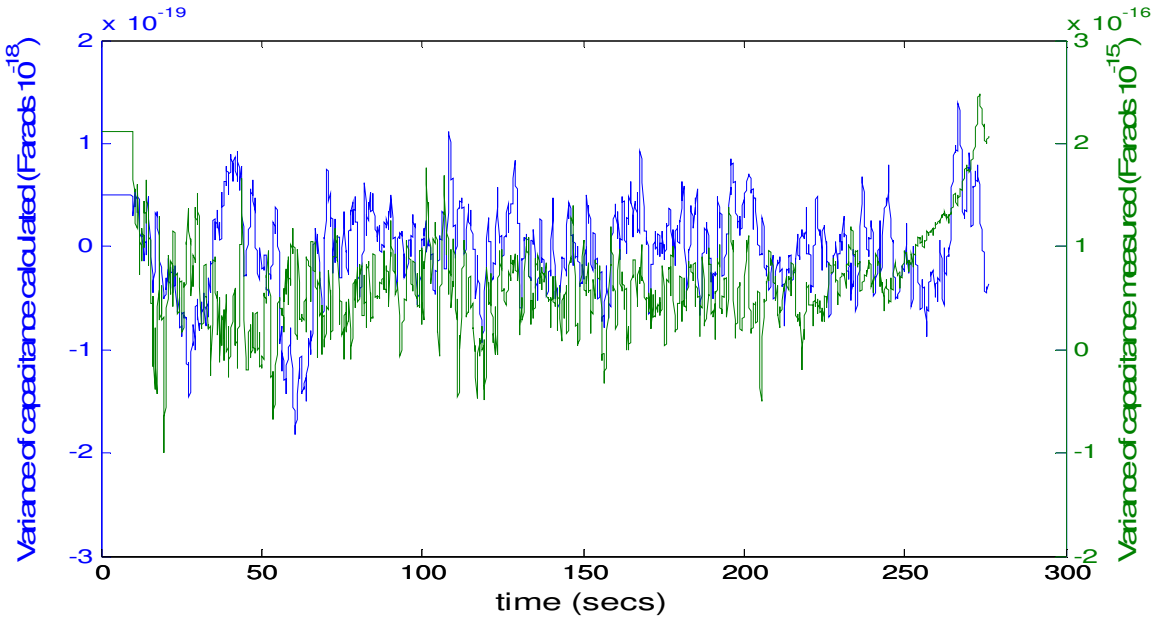


Figure 30. The 0.1 Hz capacitance signal of the total capacitance in HumiSense is shown in green. The 0.1 Hz expected total capacitance of HumiSense is shown in blue.

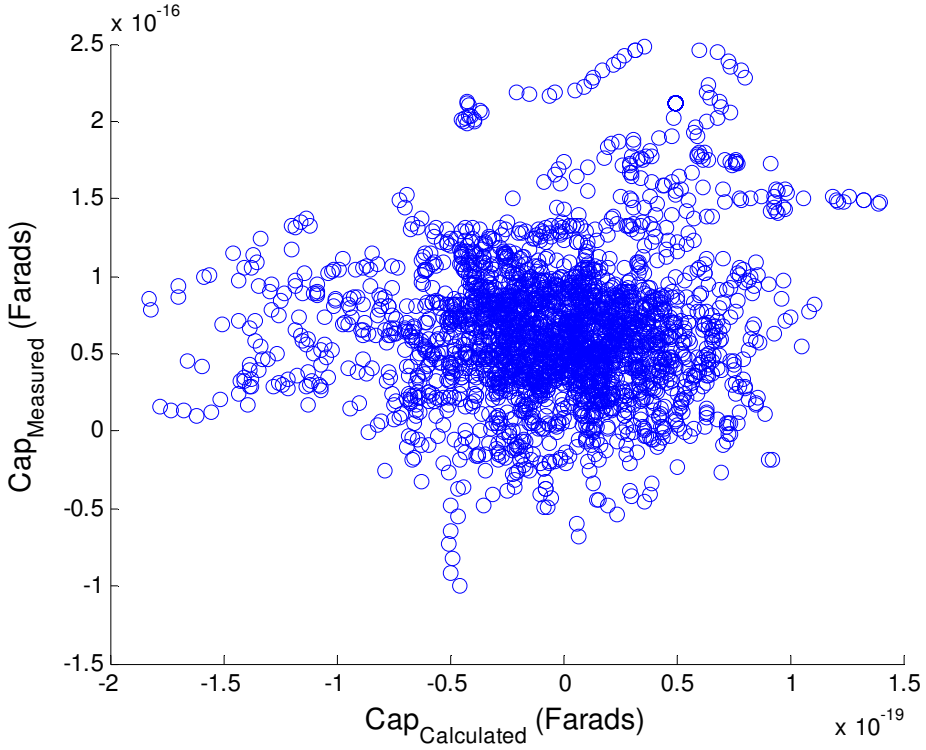


Figure 31. Correlation of capacitance signals shown in Figure 30.

Figure 32 shows the normalized power spectrum plots for the 10 Hz calculated capacitance signal and the 10 Hz measured capacitance signal. The plots are similar which potentially hints that HumiSense could be used for measuring water vapor density perturbations in the air. However, due to the lack of slope in the signal, no concrete conclusions can be made about HumiSense. Sensors that can accurately capture turbulent perturbations in the air show a slope of approximately -0.33 in their power spectrum plot in the frequency range of 0 to ½ capturing frequency.

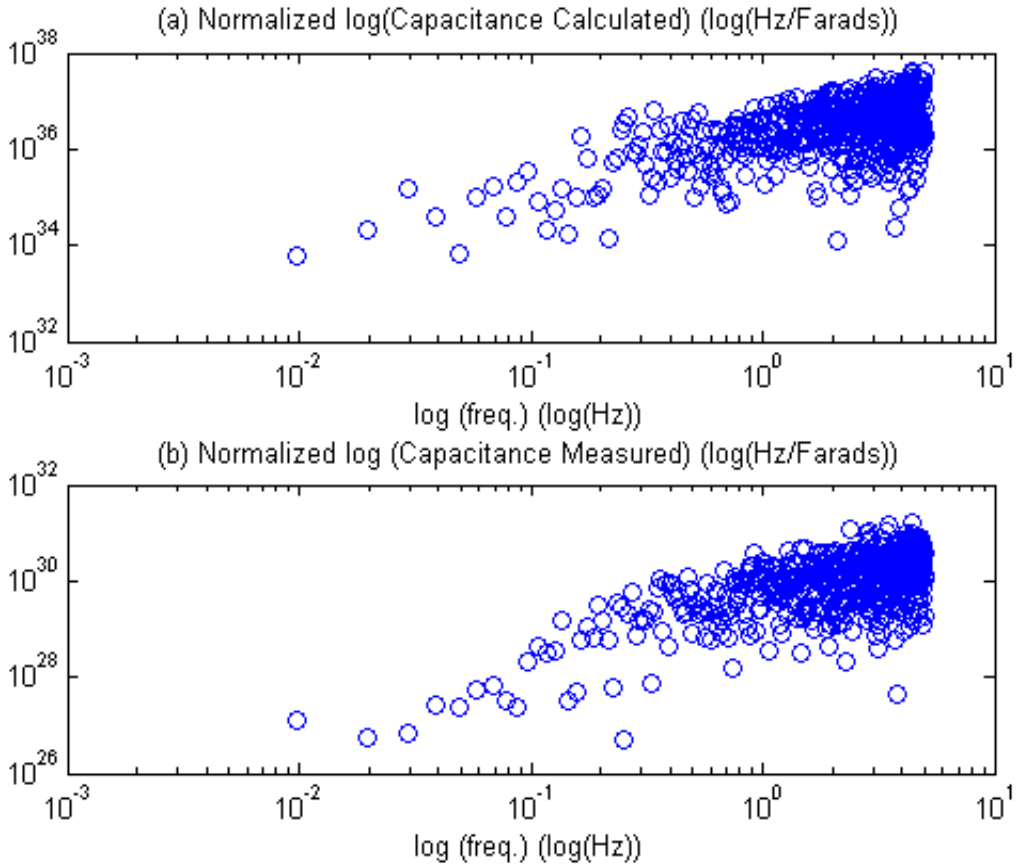


Figure 32. (a) Normalized Power Spectrum plot of the 10 Hz calculated capacitance signal. (b) Normalized Power Spectrum plot of the 10 Hz measured capacitance

signal.Figure 33 shows the normalized power cospectrum plot for the correlated signal of the 10 Hz calculated capacitance signal and the 10 Hz measured capacitance signal. Also, due to the lack of slope in the signal, no concrete concluding remarks can be made about HumiSense. As with the power spectrum plots, a slope of approximately -0.33 in the power cospectrum plot of the sensor in the frequency range of 0 to ½ capturing frequency is expected for accurate turbulent signal measuring sensor.

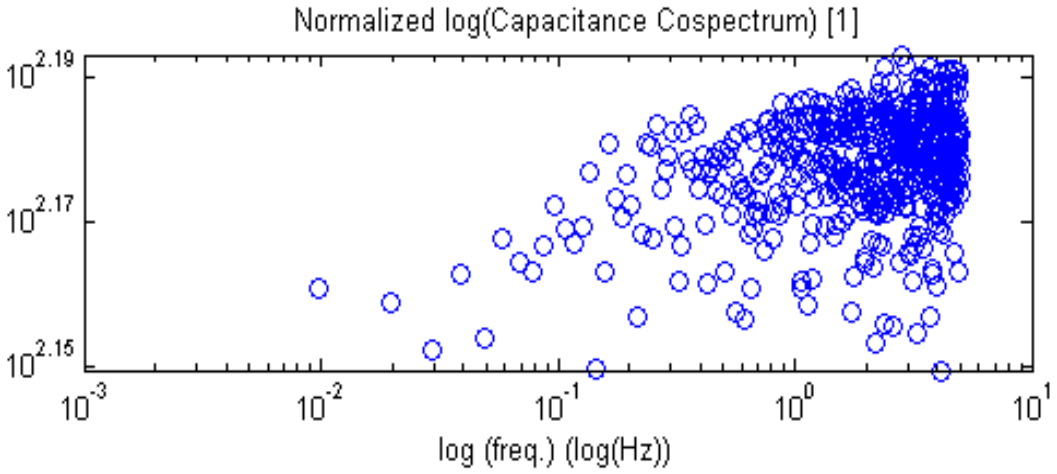


Figure 33. Normalized Power Cospectrum for the correlated signal of the 10 Hz calculated capacitance signal and the 10 Hz measured capacitance signal.

5 Discussion

5.1 Steady State Condition Experiments

The difference frequency shows discrete count levels in its noise signals. This can easily be seen in Figures 13, 14, 15, 16, 18, 19, 21, and 22. In Figures 15 and 19 there are clear discrete levels at multiples of 1kHz. In Figures 15 and 19 there are obvious +/-1kHz bands, while there are hints of quantization level of +/- 500 and +/- 250 Hz. In Figure 16, there are bands of +/- 500 and +/- 250 Hz. In Figure 22, there are also obvious +/-1 kHz bands, but also +/- 2 and +/- 3 kHz bands.

Initially, it was thought that these discretizations in the noise patterns could be due in part to 60 Hz AC power line signals. It was thought that the tip of the inner electrode of the sensor which is not shielded (with the outer electrode) was potentially having noise coupled into the oscillator circuit. There was concern about low voltage signals getting coupled and having an effect in the mixer, comparator, and/or counter. Dr. James Wagner performed two experiments in LTSpice, a Simulation Program with Integrated Circuit Emphasis (SPICE) solver software, to test the possibility of 60 Hz AC noise being coupled into the signal. In the first experiment, Dr. Wagner modeled the mixer output to see what effect noise in the mixer output would have on the comparator in the microcontroller. The noise that went into the comparator was very low in amplitude but it was seen in the output of the comparator.

Since the noisy comparator output was worrisome, Dr. Wagner performed another spice experiment were the mixer, the filter, and the comparator were modeled. He mixed two signals of 1 Volt in amplitude and 0.999 and 1 MHz in frequency respectively. A 100 Hz signal of 0.1 V was summed to the 0.999 MHz signal as the "noise" signal. Then, he modeled the exact filter that was built for HumiSense. The output of the filter was very "clean". The noisy spikes seen in the output of the comparator of experiment 1 were reduced by nine to ten orders of magnitude. Thus, the discrete noise spikes seen at +/- 0.25, 0.5, 1, and 2 kHz are highly unlikely due to 60 Hz noise coupling from AC power lines.

The distribution of the noise looks multimodal, which is very suggestive of some sort of "locking" happening. In other words, there seems to be an interaction between two components of the circuit that favors shifts of +/- 1 and 2 KHz. Locking is always a potential problem when there are analog and digital circuits in close proximity and they are not perfectly isolated from each other. It might also be noise coming from the microcontroller being pushed back into the crystal oscillator. The emitter follower and the logic buffer should provide sufficient isolation, but simple proximity might come into play here, also. Even though, there probably is some locking present in the circuit, from Figures 17, 20, and 23, we see that the "high" frequency noise spikes are not the norm but the oddity in the distributions shown by these figures. This means that the

grand majority of the noise if not made up of them. It would still be relevant to fully understand why those “high” frequency noise spikes are showing up.

5.2 Field experiments

HumiSense was deployed in the field in October 2011. Due to the unintended noise, it is hard to conclude whether the capacitance data recorded during the field test from October 6th 2011 is correlated to the expected capacitance signal calculated from the 10 Hz water vapor density signal, barometric signal, temperature signal obtained from the Licor LI-7500 and the Metek FHN-1 R.M. It is difficult to tell if there is any correlation from just looking at Figures 24, 26, 28, and 30. But, after looking at Figures 25, 27, 29, and 31 and knowing that their R^2 values are 0.0040%, 0.3782%, 0.5700%, 24.2852% respectively, there does not seem to be any useful correspondence. Lastly, not much can be concluded from Figures 32 and 33due to the lack of -1/3 slope in the normalized power spectrum and cospectrum plots of the capacitance signals.

Looking at Figure 24, the variability in the capacitance measured signal’s amplitude seems to decrease with time. This is because the difference frequency is getting smaller (approaching zero Hz) with time. Due to our period and frequency counting algorithm, the smaller the difference frequency, the smaller the counter step became.

In other words, each counting step has variability added with it, and the smaller the difference frequency was, the smaller the variability would be.

5.3 Future HumiSense generations

Our current counting algorithm, with the current components, gives us a maximum theoretical counting resolution of 48.23 Hz/count (see section 3.5.9.3). This maximum resolution is not appropriate for our high frequency water vapor density sensor. According to the calculations in section 3.5.4, to measure a change of 1ppm (which could correspond to a 1% change in humidity) in our sensor's dielectric constant, the counting resolution must at least be 2.77 Hz/count. Our current board's resolution could be improved to higher than 1ppm if we switched to the period counting algorithm and we kept the differential frequency no higher than 6.9 kHz. With the current parts in the board, given that they had varied tolerances of either 1 or 5% of their rated value, it is hard to accurately provide a differential frequency of no higher than 6.9 kHz under different weather conditions. But, other changes can be done in HumiSense to improve its sensitivity substantially.

Changing the oscillator topology of the sensing oscillator from Clapp to Hartley, even though more expensive, it would allow the creation of a custom tapped inductor that could improve the sensitivity by several fold. The Hartley oscillator would only require the sensor capacitance plus one or two capacitances to allow for tuning of the

frequencies i.e. one or two digital trimmers and air-core coiled inductors could be used to reduce stray coupling (McLucas 2006). Also, if we recall equation 28,

$$C_{Coaxial} = \frac{2\pi\varepsilon_0\varepsilon_r d}{\ln\left(\frac{b}{a}\right)} \tag{28}$$

where *d* is the length of the capacitors, *a* is the diameter of the inner electrode, and *b* is the diameter of the outer electrode, we can see that the sensitivity of HumiSense can be increased by increasing either the diameter of the inner electrode (*a*) or the length of the capacitor (*d*). Since increasing the length of the capacitor is not desirable, to improve the sensitivity the inner electrode's diameter would be increased. In the current HumiSense, the inner conductor has a diameter of 0.63cm and the outer conductor has a diameter of 11.7cm. Increasing the inner conductor's diameter to near 2.61 cm would double the value of the sensor's capacitance and hence the sensitivity would be doubled.

The sensitivity can be further improved by increasing the resonant frequency of the sensing oscillator to 32MHz. Doubling the sensing oscillator's frequency quadruples the sensitivity since it reduces the total capacitance by a factor of four. The temperature coefficient of the inductor can also be substantially reduced by going to a smaller size inductor with air core instead of a ferrite core inductor like the one currently in the device. Lastly, the software can be improved so that the trimmers are always running at the values that the sensor is most sensitive.

Another alternative would be to instead of using an analog mixer, a low pass filter and a digital counter to measure the changes in frequency between the oscillators, to use an all-digital Phase-locked loop (PLL) connected to an Analog-to-Digital converter (AD). The Phase-locked loop can detect the changes in frequency and modify the sensing oscillator so that its frequency matches the reference oscillator's frequency. The modifications done by the PLL can be measured and discretized by the AD and finally a microcontroller can log the changes in frequency by reading the AD's output.

6 Conclusion

The data recorded using the prototype of the first generation HumiSense built for this project was found to be inconclusive for measuring rapid changes in water vapor density, at least, within the materials budget and the circuit design selected. The literature seems to agree that the beat frequency method is adequate for measuring water vapor density in the air. So, improving the current HumiSense prototype might result in a working version that can measure high frequency water vapor density. Most of the publications on measuring water vapor density with the beat frequency method either used a non-polar calibration gas (Hay, Martin, and Turner *1961*, Jelatis *1948*, Tregidga *1940*) or had a "reference cavity" with dry air (Stokesberry and Hasegawa *1976*, Hasegawa and Stokesberry *1975*, Sargent *1959*, Crain *1950*). A specific publication in the literature used a different oscillator topology, the Hartley topology (Tregidga *1940*). Including a calibration gas or a reference cavity, or simply changing the topology of the oscillator might result in an improved prototype.

According to our calculations, a change of 1 count with the current HumiSense and using the frequency and period counting algorithm would correspond to a change of 25ppm in the dielectric constant off the sensor. A counting resolution of 25ppm is not adequate for measuring high frequency water vapor density fluctuations when a change of 1% relative humidity, a change of 1K, and a change of 0.0013 atm,

correspond to a change of the signal of interest by 1.0ppm, 2.0ppm, and 0.7ppm respectively. For the counting resolution of HumiSense to be at least 1ppm per count, it was calculated that the counter must be able to measure changes of 2.29 Hz per count. The only documented study using the beat frequency method with adequate resolution for measuring accurately high frequency changes in the dielectric constant of a coaxial capacitor was Jelatis (1948) study. Nonetheless, Jelatis resolution was impressive but the gas capacitor he is had a capacitance of 500pF. That capacitance is more than 100 times the capacitance of our sensor. It would not be possible under our design constraints to build a 500pF coaxial capacitor. All the other studies that employed the beat frequency method and reported their resolutions, did not get high enough resolutions to be used for measuring high frequency changes in the dielectric constant of a coaxial capacitor: Tregidga (1940), Crain (1948), and Sargent (1959) reported resolutions of 200ppm, 10.7 to 21.4ppm (depending on the gas), and 56.1ppm.

Several improvements may be made to the resolution and accuracy of HumiSense. First, the number of parts needs to be reduced. This will reduce the cumulative variance in temperature coefficients, and in effect, reduce the overall temperature dependence. Also, components with a tighter and lower temperature coefficient may be used. The inductor is one of the parts with larger temperature coefficient. Replacing the ferrite core inductor with an air core inductor would reduce the temperature coefficient of the inductor significantly. This will not only lower the

temperature dependence, but also create an opportunity to add a statistically calculated offset to the measurements, since the total temperature offset will have a lower cumulative variance. As explained in the Discussion chapter, the resolution of HumiSense could be improved by changing the counting algorithm to period counting, by increasing the resonant frequency of the sensing oscillator, by increasing the diameter of the inner electrode, or by switching to a different oscillator topology.

The resolution of the sensing capacitor in HumiSense could also be improved by changing the oscillator topology to either Hartley or Colpitts. The Hartley topology would require less parts and the tuning of the circuit could potentially be done with the inductors. The Colpitts topology would allow for the sensing capacitor to have a more significant effect on the frequency of the oscillator. Lastly, adding a reference cavity or using a calibration gas could also be considered for the implementation of an improved HumiSense. Either of the two should allow for the qualification of HumiSense. Lastly, the heterodyning of the signals could be done with a different electronic device, an all-digital Phase-locked loop (PLL). With the use of an all-digital Phase-locked loop (PLL) and an Analog-to-Digital converter (AD), changes in frequency would be detected by the PLL and then, the PLL would modify the sensing oscillator by countering for the changes in frequency detected till the sensing frequency matched the reference frequency. Then, the AD in the microcontroller would discretize the shifts in frequency and finally the microcontroller could log the AD activity.

Bibliography

(2001) Guideline on the Use of Fundamental Physical Constants and Basic Constants of Water. Gaithersburg, Maryland, USA, The International Association for the Properties of Water and Steam.

ALFONSI, G., 2009. Reynolds-Averaged Navier–Stokes Equations for Turbulence Modeling. *Applied Mechanics Reviews*, 62(4), p.040802.

BEAN, B. R., (1962) The Radio Refractive Index of Air. *Proceedings of the IRE* 50, no. 3, 260-273.

BERTONE, G. A., MEIKSIN, Z. H. & CARROLL, N. L. (1991) Elimination of the anomalous humidity effect in precision capacitance based transducers. Instrumentation and Measurement, IEEE Transactions on, 40, 897-901.

BIRNBAUM, G. (1950) A Recording Microwave Refractometer. Review of Scientific Instruments, 21, 169-176.

BIRNBAUM, G. & CHATTERJEE, S. K. (1952) The Dielectric Constant of Water Vapor in the Microwave Region. Journal of Applied Physics, 23, 220-223.

BISHOP, D. M. & CHEUNG, L. M. (1982) Vibrational Contributions to Molecular Dipole Polarizabilities. Journal of Physical and Chemical Reference Data, 11, 119-133.

BRUTSAERT, W. (1982). Evaporation into the Atmosphere: Theory, History, and Applications, Springer.

CARR, J.J. (1993). Mastering Oscillator Circuits Through Projects and Experiments, McGraw-Hill Companies, The.

CLAPP, J. K. (1948). An Inductance-Capacitance Oscillator of Unusual Frequency Stability. Proceedings of the IRE, 36 (3), 356– 358.

CRAIN, C. M. (1948) The Dielectric Constant of Several Gases at a Wave-Length of 3.2 Centimeters. Physical Review, 74, 691.

CRAIN, C. M. (1950) Apparatus for Recording Fluctuations in the Refractive Index of the Atmosphere at 3.2 Centimeters Wave-Length. Review of Scientific Instruments, 21, 456-457.

DIAS, N. L., DUARTE, H. F., MAGGIOTTO, S. R. & DIO GRODZKI, L. (2007) An attenuated eddy covariance method for latent heat flux measurements. Water Resources Research, 43, 1-9.

ENGLUND, C. R. CRAWFORD, A. B. AND MUMFORD, W. W. (1935) Further results of a Study of Ultra-Short-Wave Transmission Phenomena. The Bell System Technical Journal, XIV, 369 - 387.

FORD, L. H. (1948) The effect of humidityon the calibration of precision air capacitors. Measurement Sections 13-16.

HASEGAWA, S. & STOKESBERRY, D. P. (1975) Automatic Digital Microwave Hygrometer. Review of Scientific Instruments, 46, 867-873.

HAY, D. R., MARTIN, H. C. & TURNER, H. E. (1961) Light-Weight Refractometer. The Review of Scientific Instruments, 32, 693-697.

HECTOR, L. G. & WOERNLEY, D. L. (1946) The Dielectric Constants of Eight Gases. Physical Review, 69, 101.

HUGHES, J.V. & ARMSTRONG, H.L. (1951) Dielectric Constant of Dry Air. Journal of Applied Physics, 23, N°5, 501-504.

HYSON, P. & HICKS, B. B. (1975) Single-Beam Infrared Hygrometer for Evaporation Measurement. Journal of Applied Meteorology, 14, 301-307.

JELATIS, J. G. (1948) Measurements of Dielectric Constant and Dipole Moment of Gases by the Beat-Frequency Method. Journal of Applied Physics, 19, 419-425.

KAYE, G.W.. & LABY, T.H. (1995) Tables of physical and chemical constants, Longmans.

KERR, D. E. (1951) Propagation of Short Radio Waves, McGraw-Hill Book Company, Inc.

KIMOTO, A. & SHIDA, K. (1996) Estimation of position and relative dielectric constant of dielectric in a cylindrical air condenser by new capacitance measurement method. Japanese Journal of Applied Physics Part 1-Regular Papers Short Notes & Review Papers, 35, 5896-5900.

MCLUCAS, J. (2006) Hartley Oscillator Requires No Coupled Inductors. *EDN* 51 (22) (October 26): 112–114.

LEA, N. (1945) Notes on the stability of LC oscillators. Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution of, 92, 261-274.

LONG, R. A. K. (1953) The effect of humidity variation on the operation of an electronic proximity switch. Journal of Scientific Instruments, 30, 422-424.

LORRAIN, P., and D. R Corson. (1990). Electromagnetism: principles and applications. W.H. Freeman.

LOWE, P.R. (1977) An Approximating Polynomial for the Computation of Saturation Vapor Pressure. *Journal of Applied Meteorology*, 16(1), pp.100-103.

NURGE, M. A. & PERUSICH, S. A. (2010) In-line capacitance sensor for real-time water absorption measurements. Sensors and Actuators B-Chemical, 150, 105-111.

ONG, J. B., YOU, Z. P., MILLS-BEALE, J., TAN, E. L., PERELES, B. D. & ONG, K. G. (2008) A Wireless, Passive Embedded Sensor for Real-Time Monitoring of Water Content in Civil Engineering Materials. Ieee Sensors Journal, 8, 2053-2058.

ONG, K. G., GRIMES, C. A., ROBBINS, C. L. & SINGH, R. S. (2001) Design and application of a wireless, passive, resonant-circuit environmental monitoring sensor. Sensors and Actuators a-Physical, 93, 33-43.

ONG, K. G. & GRIMES, C. A. (2000) A resonant printed-circuit sensor for remote query monitoring of environmental parameters. Smart Material Structures, 421-428.

PHILLIPS, W. E. (1949) The Dielectric constant of Air at a Wavelength of 10 Centimeters. Proceedings of the I.R.E.

RIETVELD, G. & VAN DEN BROM, H. E. (2009) DC and Low-Frequency Humidity Dependence of a 20 pF Air-Gap Capacitor. Ieee Transactions on Instrumentation and Measurement, 58, 967-972.

ROBERTS, L. (2011) 9 Billion? Science 333, 6042 (July 29), 540 -543.

SALZMAN, W. R. (2001). Clapeyron and Clausius–Clapeyron Equations. Chemical Thermodynamics. University of Arizona.

SARGENT, J. (1959) Recording Microwave Hygrometer. Review of Scientific Instruments, 30, 348-355.
SCANLON , T.M. & ALBERTSON, J.D. (20012) Turbulent transport of carbon dioxide and water vapor within a vegetation canopy during unstable conditions: Identification of episodes using wavelet analysis. Journal of Geophysical Research, 106, 7251–7262.

SHEN, L.C. & KONG, J.A. (1983) Applied electromagnetism, Brooks/Cole Engineering Division.

SMITH, E. K. & WEINTRAUB, S. (1953) The Constants in the Equation for Atmospheric Refractive Index at Radio Frequencies. Proceedings of the IRE, 41, 1035-1037.

STOKESBERRY, D. P. & HASEGAWA, S. (1976) Automatic Digital Microwave Hygrometer, Model 2. Review of Scientific Instruments, 47, 556-558.

STRANATHAN, J. D. (1935) Dielectric Constant of Water Vapor. Physical Review, 48, 538-544.

SWINBANK, W. C. (1951) The Measurement of Vertical Transfer of Heat and Water Vapor by Eddies in the Lower Atmosphere. Journal of Atmospheric Sciences, vol. 8, Issue 3, pp.135-145.

TAYLOR, G.I. (1938) The Spectrum of Turbulence. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 164(919), pp.476–490.

TREGIDGA, A. C. (1940) The Dielectric Constant of Water Vapor at a Frequency of 42 Megacycles. Physical Review, 57, 294-297.

TSAMIS, E. D. & AVARITSIOTIS, J. N. (2005) Design of planar capacitive type sensor for "water content" monitoring in a production line. Sensors and Actuators a-Physical, 118, 202-211.

VETROV, V. V. & KATUSHKIN, V.P. (1974) Analysis of Capacitive Measurements of Relative Air Humidity. Measurement Techniques, 17, 750-752.

WEBB, P. & NEUGEBAUER, M. K. (1954) Recording Dielectric Hygrometer for Expired Air. The Review of Scientific Instruments, 25, 1212-1217.

ZAHN, C. T. (1924) The Electric Moment of Gaseous Molecules of Halogen Hydrides. Physical Review, 24, 400.

ZAMBOV, L. M., POPOV, C., PLASS, M. F., BOCK, A., JELINEK, M., LANCOK, J., MASSELI, K. & KULISCH, W. (2000) Capacitance humidity sensor with carbon nitride detecting element. Applied Physics A: Materials Science & Processing, 70, 603-606.

Appendices

Appendix 1. Alternative sensing capacitor

An alternative option for a sensing capacitor would be to build a *printed board LC resonant circuit* like the resonant printed-circuit sensor built by Ong and Grimes (2000). Instead of having a resonant circuit made up with a coaxial or parallel plate capacitor built of metal mesh and a chip inductor, the sensor capacitor and the inductor can be etched directly on to the circuit board. In Ong and Grimes (2000) LC circuit, the capacitor is an interdigital capacitor that is surrounded by a spiral inductor.

Appendix 2. C code

```
//
*********************************************************************************
*********
// Timer Compare Match program for HumiSense board V2
// Measures main clock events inside the filtered (LPF) mixed period length
//
// Device     : ATmega168xx where xx = nothing, A, P, PA
//
// Port Pin Assignments
//   PORTB    PB0 EN1              digital cap 1 enable
//                  PB1    DAT           digital cap data line
//                  PB2 EN0              digital cap 0 enable
//                  PB3 MOSI      serial programming output
//                  PB4 MISO      serial programming input
//                  PB5 SCK              serial programming clock
//                  PB6    XTAL1  external ref osc input
//                  PB7 not used
//   PORTC    PC0 not used
//                  PC1 not used
//                  PC2 not used
//                  PC3 ADC3      thermistor analog input
//                  PC4 not used
//                  PC5 not used
//                  PC6 /RESET
//                  PC7 not implemented
//   PORTD    PD0 RXD              RS232 input
//                  PD1 TXD              RS232 output
//                  PD2 GIO              general purpose I/O
//                  PD3 GIO              general purpose I/O
//                  PD4 GIO              general purpose I/O
//                  PD5 T1        frequency input
//                  PD6 AIN0      input = mixer output
//                  PD7 AIN1      input = mixer output
//
// Fuse Settings (nb: Fuse bytes ARE different between Mega168 & Mega320)
//                           (    listed settings are for Mega168)
//  EXTENDED  BOOTSZ1       prog   default       minimum boot section
//                          BOOTSZ0 prog  default       /
//                          BOOTRST       unprog default              standard
reset vector
//
//     HIGH          RSTDBL unprog default              normal RESET function
//                          DWEN   prog   NOT DEFAULT   enable debug
//                          SPIEN  prog   default              enable serial
programming
//                          WDTON  unprog default              watchdog NOT
always on
//                          EESAVE  unprog       default              don't
preserve EEPROM when programming
//                          BODLEVEL2 unprog default
```

```
//                          BODLEVEL1 unprog default |  do not enable brown out
detection
//                          BODLEVEL0 unprog defalut /
//
//  LOW                     CKDIV8 unprog NOT DEFAULT          do not divide
clock by 8
//                          CKOUT  unprog default                     no clock
output
//                          SUT1   unprog default        fastest startup for
spec'd clock
//                          SUT0   prog   default        /
//                          CKSEL3 prog   default
//                          CKSEL2 prog   default     | external clock input 0-
20MHz
//                          CKSEL1 prog   NOT DEFAULT |
//                          CKSEL0 prog   default          /
//
// This code is driven by a state machine. The state variable is count_status.
The
// states are listed next (names prefixed by "k" to denote "konstant").
//
// State = kOriginal The measurement process is the old, pure period method used
to
//                                       tune the digital capacitors. It stays
in this state though-out
//                                       multiple measurements.
//
// State = kWait           The state between the end of computation and the
start of a
//                                       new measurement. While in this state,
the input capture ISR updates
//                                       LastValue on each interrupt so that
there will be a valid
//                                       LastValue when the next measurement
cycle begins, but that is all
//                                       the ISR does.In this state, 25ms/40Hz
timer roll-overs are
//                                       counted. When the appropriate count is
rached, a transition
//                                       to state = kMeasure takes place.
//
// State = kMeasure        This state is triggered (transition out of kWait) by
the
//                                       25ms/40Hz timer. The state transition
triggers the transmission
//                                       of the most recent data, starts the
ADC, and starts the
//                                       period measurement. While in this
state, capture interrupts
//                                       are counted, timer count totals are
generated, and the
//                                       timer count total is compared against
the maximum allowed
```

```
//                                        count threshold. If the threshold is
exceeded, state changes
//                                        to kCompute. Also, while in this
state, 25ms/40Hz timer
//                                        roll-overs are counted.
//
// State = kCompute        This state is triggered by the exit from kMeasure.
Necessary
//                                        computations and string constructions
are carried out. Also,
//                                        25ms/40Hz timer roll-overs are
counted. This state is exited
//                                        when computation is complete; the
transition is to kWait.
//                                        While in this state, the input capture
ISR updates
//                                        LastValue on each interrupt so that
there will be a valid
//                                        LastValue when the next measurement
cycle begins, but that is all
//                                        the ISR does.

// July 29th 2011
// Last revised: Dec 7th 2011
// This is the same as Test_Humisense_PeriodM_v9 just with a slower transmit
rate
//
********************************************************************************
*********

//Define the Clock
#define F_CPU 16000000UL //16 MHz

//Include System Header Files
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h> //*JDW4 exposes _delay_us() and _delay_ms()
#include <stdlib.h>         //*JDW4 exposes itoa()
#include <string.h>         //+JDW4 exposes strcat()

//Include UserDefined Header Files and Dependencies
//#include "SimpleDefs.h"

//Definitions
#define DAT PB1
#define EN0 PB2
#define EN1 PB0
#define CAPPORT PORTB
#define IO1 PD4
#define IO2 PD3
#define IO3 PD2
#define IOPORT PORTD
```

```
//#define ModeOrig 0x00                         //-JDW3 +JDW orginal period
determination rules
//#define ModeNew     0xff                //-JDW3 +JDW new period determination
rules
//#define CountRun  0x00                  //-JDW3 +JDW measurement running
//#define CountDone 0x02                  //-JDW3 +JDW measurement finished
//#define CountPause 0x01                 //-JDW3 +JDW count done, wait for
trigger to begin again

//state variables for count_status
#define kOriginal    0x00                 //+JDW3 see state machine description,
above
#define kWait        0x01                 //+JDW3 ditto
#define kMeasure     0x02                 //*JDW4 +JDW3 ditto (JDW4 error -
duplicated kWait!)
#define kCompute     0x03                 //*JDW4 +JDW3 ditto

//ADC channel addresses
#define kADC_Chan3   0x03                 //+JDW3
#define kADC_clear   0xf0                 //+JDW3 clears the MUX3:0 bits

#define EN0LO  CAPPORT &=~(1<<EN0) //trimmer 0 is disabled
#define EN0HI  CAPPORT |=(1<<EN0)  //trimmer 0 is enabled
#define EN1LO  CAPPORT &=~(1<<EN1) //trimmer 1 is disabled
#define EN1HI  CAPPORT |=(1<<EN1)  //trimmer 1 is enabled
#define DATLO  CAPPORT &=~(1<<DAT) //DAT is low
#define DATHI  CAPPORT |=(1<<DAT)  //DAT is high
#define IDLE   CAPPORT ~((1<<DAT)|(1<<EN0)|(1<<EN1))

//Global Variables
volatile uint32_t avevalue_counts  = 0x00;
volatile uint32_t FinalValue              = 0x00;       //*JDW was uint16_t
volatile uint32_t FinalCount              = 0x00;              //+JDW
volatile uint16_t LastValue               = 0x00;
volatile uint8_t  SampleCount                    = 0x00;              //+JDW
software counter for 10Hz sample rate
volatile uint32_t AveValue;                              //+JDW7 was local to
ISR(TIMER1_CAPT_vect)
volatile uint16_t Difference;
static uint16_t index0                           = 0x00;
static uint16_t index1                           = 0x00;
static uint16_t kCountMin                 = 0x2A;
static uint16_t kCountMax                 = 0x04DD;
char* package                                    = 0x00;       //*JDW surrogate
ptr to buffer
char buffer[20] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00}; //First message transmitted ="00"
char shortbuf[14];
char* returnASCII                                        = 0x00;       //+JDW
surrogate ptr to shortbuf

//volatile uint8_t mode                          = ModeOrig;   //-JDW3 +JDW
```

```
volatile uint8_t count_status            = kOriginal;//*JDW3 +JDW

//*********************************************************************
//                            INPUT CAPTURE ISR
//
//*********************************************************************

//The Interrupt Service Routine
ISR(TIMER1_CAPT_vect){
                    uint16_t ThisValue;
                    //uint16_t Difference;

      //Update
      ThisValue           =ICR1;

      //Calculate the Difference
      if (ThisValue > LastValue){
            Difference    = ThisValue - LastValue;
      }
      else{
            Difference    = (0xFFFF - LastValue) + ThisValue;
      }
   LastValue       = ThisValue;

      //+JDW3 start of state-dependent code
      if (count_status == kOriginal){    //*JDW3 +JDW
            //Calculate SUM, then after 256 samples, calculate the average
            if (avevalue_counts <= 0xFF){
                  AveValue = AveValue + Difference;               //+JDW7
                  avevalue_counts++;
            }
            else{
                  //Obtaining the average
                  FinalValue                 = (uint32_t)(AveValue >> 8);
      //Integer Division, Base-2 +JDW7 //EE 12/9/2011
                  //Resetting values
                  avevalue_counts            = 0x00;
                  AveValue = 0;                                   //+JDW7
                  }
            }//Stef-->I think it was missing +JDW3 YES!
      else if (count_status == kMeasure){ //*JDW3 +JDW

//*********************************Diagnostic*****************************
************
            if ( TIFR2 & (1<<OCF2A) ) { //*JDW6

                  TIFR2 |= (1<<OCF2A); //clear Timer2 OC2A match flag

                  SampleCount ++;                  //increment sample count
every 1ms

            } //end if OCF2A set in TIFR2
```

```
//*************************************************************************
*************

              //+JDW this entire block
              //test against new algorithm rules
              //Calculate SUM, then after sum reaches 0x00051A81, reset and
start over
              AveValue = AveValue + Difference;
              if ((AveValue & 0x14C000) == 0x14C000){                //+JDW7
ditto
                     FinalValue                 = AveValue;          //+JDW7
                     FinalCount                 = avevalue_counts;
                     count_status         = kCompute;          //*JDW3 +JDW
       //done with measure. Now compute
                     //Resetting values
                     avevalue_counts            = 0;
                     AveValue                   = 0;          //+JDW7
                     }
       else
                     avevalue_counts++;
       } //+JDW
       //+JDW3 end state dependent block
} //end ISR

//*************************************************************************
**********

//intialize
void intialize(void){

       //                                    PORTS

       //Settings directions: EN1 as output, DAT as output, EN0 as output, MOSI
as input, MISO as output, SCK as input, MICRO_CLK as input
       DDRB = (1<<DDB2)|(1<<DDB1)|(1<<DDB0);

       //Pull-up resistors for inputs in PORTB
       //+JDW4 this also sets CAPPORT.DAT, CAPPORT.ENO and CAPPORT.EN1 low, as
required for initial entry into either CAPDATn() function
       PORTB = (1<<PORTB5)|(1<<PORTB4)|(1<<PORTB3);

       //Settings directions:  Thermistor as input, RESET as input
       DDRC = 0x00;

       //Settings directions: RXD as input, TXD as output, TMS as input, TDO as
output, TCK as input, BeatSignals as inputs
       //+JDW5 PD4 = IO1 as diagnostic output
       DDRD = (1<<DDD4)|(1<<DDD3)|(1<<DDD2)|(1<<DDD1);

       //Pull-up resistors for inputs in PORTD
       PORTD = (1<<PORTD3);

       //Disabling the Digital Input Buffers
```

```
	//+JDW4 NB: Lets ONLY disable actual ADC inputs
	//-JDW4 DIDR0 |=
(1<<ADC5D)|(1<<ADC4D)|(1<<ADC3D)|(1<<ADC2D)|(1<<ADC1D)|(1<<ADC0D);
	DIDR0 |= (1<<ADC3D); //+JDW4
	DIDR1 |= (1<<AIN1D)|(1<<AIN0D);

	//                                ANALOG COMPARATOR

	//Setting the Analog Comparator Status Register (ACSR)
	ACSR = (1<<ACIC);

	//                                              TIMER1 - 16-BIT INPUT
CAPTURE

	//Enabling the Input Capture Interrupt
	TIMSK1 = (1<<ICIE1);

	//Setting the Timer Counter Control Register 1 A
	TCCR1A = 0x00;

	//Setting the Timer Counter Control Register 1 B
	//The Noise Canceller is OFF; must set bit 7 for it to turn ON
	TCCR1B = (1<<ICES1)|(1<<CS10);

	//Setting the Timer Counter Control Register 1 C
	TCCR1C = 0x00;

	//                                        TIMER2 - 8-BIT 100MS GENERATOR

	//MODE2 CTC, OCR2A compare, no output
	TCCR2A = (1<<WGM21); //*JDW4 +JDW
	//TCCR2A = (1<<WGM1); //-JDW4 Stef: I believe this is what you meant to
do 9-13-11

	//Prescale by 128 for 125KHz clock
	TCCR2B = (1<<CS22)|(1<<CS20);            //+JDW

	//Count by 125 for 1KHz ticks
	OCR2A = 125;               //*JDW3 +JDW

	//+JDW NB: rollover detection will be handled by polling to prevent
	//interrupt conflicts with Timer1 Input Capture, which MUST be
	//served promptly (and potentially at fairly high rates). That poll
	//will look at OCF2A in TIFR2.

	//                                                    ADC

	//Setting the ADC multiplexer Register (ADMUX)
	//ADMUX |= (1<<REFS1)|(1<<MUX1)|(1<<MUX0);      -JDW3
	ADMUX = (1<<MUX1)|(1<<MUX0)|(1<<REFS0);               //+JDW3 Chan 3,
Vref = Vcc, result right adjusted

	//Setting the ADC control and status Register A (ADCSRA)
```

```
	//+JDW3 turn on ADC, don't start (yet), no auto trigger, no int (polled)
	//+JDW3 prescaler must be chosen for 50KHz to 200KHz conversion clock
from
	//+JDW3 from 16 MHz F_CPU. Needs a prescale of 128 (125KHz) to 256
(62.5KHz)
	//+JDW3 Only 128 is available.
	ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);    //*JDW3

	//turn off the digital I/O on input channel

}

// Initializing USART
void InitUSART(unsigned int baudrate)
{
	// Set the baud rate
	UBRR0 = baudrate;

	// Enable USART transmitter
	UCSR0B = (1 << TXEN0);

	// set to 8 data bits, 1 stop bit
	UCSR0C = (1 << UCSZ01) | (1 << UCSZ00);
}

//Must enter IDLE state with DAT, EN0, and EN1 all low!
void CapDat0(uint16_t NumCounts){
	uint16_t TheCount = NumCounts;//should compile to an r-register

	EN0HI;               //enable trimmer 0
	while (TheCount>0){
		DATHI;           //DAT goes high
		DATLO;           //DAT goes low
		TheCount--;
	}
	EN0LO;               //disable trimmer 0
}

void CapDat1(uint16_t NumCounts){
	uint16_t TheCount = NumCounts;//should compile to an r-register

	EN1HI;               //enable trimmer 1
	while (TheCount>0){
		DATHI;           //DAT goes high
		DATLO;           //DAT goes low
		TheCount--;
	}
	EN1LO;                   //disable trimmer 1
}

//USART transmission subroutine
void TransmitByte (char contents)
{
```

```
  // Wait for empty transmit buffer
       while ((UCSR0A & (1 << UDRE0)) == 0) { //*JDW5
              //+JDW5 following block added to update SampleCount while data is
transmitted
              if ( count_status != kOriginal) {

                     if ( TIFR2 & (1<<OCF2A) ) { //*JDW6

                            TIFR2 |= (1<<OCF2A); //clear Timer2 OC2A match flag

                            SampleCount ++;                    //increment
sample count every 1ms

                     } //end if OCF2A set in TIFR2
              } //end if count_status != kOriginal
       }  //+JDW5

  // Start transmission
       UDR0 = contents;
} //+JDW5 end TransmitByte

//Transmits a entire ASCII string across the UART
//Input Arguments:
//               *data: A pointer to the first location of the string.
void TransmitASCII(char *data){
       while (*data){//Loop until data == '\0', end of string
              TransmitByte(*data);   //transmit the contents at pointer address
              data++;                            //increment to the next pointer
       }
       TransmitByte(0x0D);        //Transmit Line feed character
}

void delay_us(uint16_t us) {
       while (us) {
              _delay_us(1);
              us--;
       }
}

void delay_ms(uint16_t ms) {
       while (ms) {
              _delay_ms(1);
              ms--;
       }
}

//Tuning of coarse trimmer
void tuneCoarseTrimmer(void){
       uint8_t NotDone;

       //Initialize flag
       NotDone = 0xFF;
```

```
                while (NotDone) {

                delay_ms(100);

                if(index1 >= 0x1F){
                        index1 = 0x00;
                        CapDat1(index1);
                }

                if (avevalue_counts > 0x02){

              if (FinalValue >250)
                  NotDone = 0x00;
                        else {
                index1++;
                        CapDat1(index1);
                        }
                }

                else {
                        index1++;
                        CapDat1(index1);
                }

        }
}

//Tuning of fine trimmer
void tuneFineTrimmer(void){
                        uint8_t             NotDone;
        static uint16_t      check_i0      = 0xFF;  //Initialize to something not
being used by index0

        //Initialize flag
        NotDone = 0xFF;

        while (NotDone) {

                delay_ms(100);

                if(index0 >= 0x1F){
                        index0 = 0x00;
                        CapDat0(index0);
                }

                if (avevalue_counts > 0x02) {


                        if (FinalValue >= 600) // Mixed frequency
                                NotDone = 0x00; //make it false

                        else if (FinalValue < 250) { // Mixed frequency
                                check_i0 = index0;
```

```
                        if (index0 >= 0x01){
                                index0--;
                                }
                        CapDat0(index0);
                }

                else {
                        index0++;
                        CapDat0(index0);
                        if (check_i0 == index0){
                                if (index1 >= 0x01){
                                        index1--;
                                        }
                                CapDat1(index1);
                        }
                }

        } //end if avevalue_counts > 2

    } //end while
}

//subroutine to make the ASCII message with the caps indexes

void makeTuneMessage(void){
    //Packaged value of CapDat Indexes
    package = ltoa(index0, buffer, 10);

    //Append the null-terminated string that contains one
    //comma to the end of the temperature string.
    package = strcat(buffer,",");

    //Convert index1 to ASCII, base 16(hex)
    returnASCII = ltoa(index1, shortbuf, 10);// ** JDW changed argument from
buffer to shortbuf

    //append period value to transmit package
    package = strcat(buffer,returnASCII);

    //TransmitByte (num);
    TransmitASCII(package);
}

// ********************************************
// MAIN PROGRAM
// ********************************************
int main(void) {
    //Local variables
    static uint16_t localindex0                 = 0;
    static uint16_t localindex1                 = 0;
    static uint32_t localFinalValue             = 0;   //*JDW was
uint16_t
    static uint32_t localFinalCount             = 0;
```

```
static uint32_t Temp                            = 0x00;


intialize();
InitUSART(103) ;     // Initialize USART and setting baudrate at 9600

//Setting up the digital trimmers
CapDat0(index0);
CapDat1(index1);

sei();

//+JDW4 Interrupts will have happened by now so clear any pending
TIFR1 |= (1<<ICF1);         //+JDW4

//Reset the input capture counter
FinalValue                  = 0;   //+JDW
avevalue_counts             = 0x00; //+JDW
//AveValue                  = 0;   //-JDW3 +JDW

//Coarse tune loop
//Loop to find the right CapDat1 value
tuneCoarseTrimmer();

delay_ms(100);

//Fine tune loop
//Loop to find the right CapDat0 value
tuneFineTrimmer();

delay_ms(100);

//Send a message with the value of the caps indexes
makeTuneMessage();

//Reset the 40Hz counter
TCNT2 = 0;                          //+JDW
TIFR2 |= (1<<OCF2A); //+JDW reset compare match flag
//+JDW NB: SampleCount = 0 since it has not yet been touched

//Reset the input capture counter
//FinalValue                = 0;   //+JDW
//avevalue_counts           = 0x00; //+JDW
//AveValue                  = 0;   //-JDW3 +JDW

//change input capture mode to new style
//mode = ModeNew;                   //-JDW3 +JDW
//wait for Timer2 trigger to begin new measurement
count_status = kWait;       //+JDW
// ********************************************
// MAIN LOOP
// ********************************************
while(1) {
```

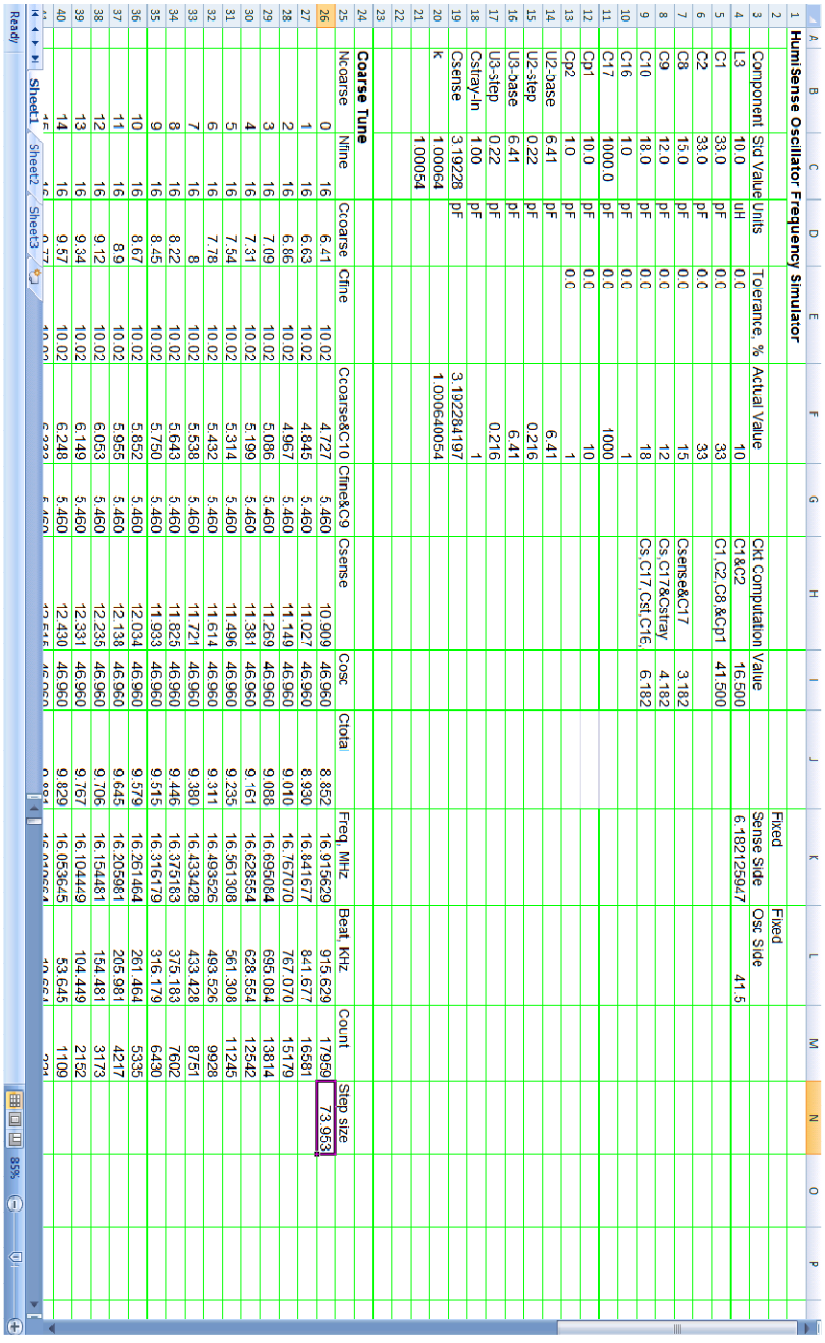
```
                buffer[0-20] = 0x00;

                if (count_status == kCompute) {    //*JDW3 +JDW

                        buffer[0-20] = 0x00;

                        //IOPORT |= (1<<IO3);       //+JDW5 sets io pin at start of
string construct DIAGNOSTIC

                        //TransmitByte (num);
                        //TransmitASCII(returnASCII);
                        localFinalValue = FinalValue;
                        localFinalCount = FinalCount;

                        //Storing the ADC conversion value
                        //Temp = ADC;

//*********************************Diagnostic*****************************
************
                        if ( TIFR2 & (1<<OCF2A) ) { //*JDW6+JDW3 Counter is reset
every 1ms (1kHz clock rate)

                                TIFR2 |= (1<<OCF2A); //+JDW3 reset the overflow flag

                                SampleCount ++;                    //+JDW3 increment
sample count every 1ms

                        } //end of TIFR2 & (1<<OCF2A)
//*************************************************************************
************

                        //-JDW4 - now started when a new measurment starts, later
in main()
                        //Start new ADC conversion: changed on 9/14/2011 EE
                        //ADCSRA &= kADC_clear;     //clear the channel selection
                        //ADCSRA |= (1<<ADSC)|kADC_Chan3;  //start the conversion

                        //Packaged value of counts and temp. temp is the leading
                        //part of the final package, so put it in there, right
                        //at the start.
                        package = ltoa(Temp, buffer, 10);

                        //Append the null-terminated string that contains one
                        //comma to the end of the temperature string.
                        package = strcat(buffer,",");

//*********************************Diagnostic*****************************
************
                        if ( TIFR2 & (1<<OCF2A) ) { //*JDW6+JDW3 Counter is reset
every 1ms (1kHz clock rate)

                                TIFR2 |= (1<<OCF2A); //+JDW3 reset the overflow flag
```

```
                              SampleCount ++;                    //+JDW3 increment
sample count every 1ms

                        } //end of TIFR2 & (1<<OCF2A)
//*************************************************************************
*************

                        //Convert FinalValue into a hex string
                        returnASCII = ltoa(localFinalValue, shortbuf, 10);  // **
JDW

                        //append period value to transmit package
                        package = strcat(buffer,returnASCII);

//*********************************Diagnostic*****************************
*************
                        if ( TIFR2 & (1<<OCF2A) ) { //*JDW6+JDW3 Counter is reset
every 1ms (1kHz clock rate)

                              TIFR2 |= (1<<OCF2A); //+JDW3 reset the overflow flag

                              SampleCount ++;                    //+JDW3 increment
sample count every 1ms

                        } //end of TIFR2 & (1<<OCF2A)
//*************************************************************************
*************

                        //Append the null-terminated string that contains one
                        //comma to the end of the temperature string.
                        package = strcat(buffer,",");              // ** JDW

                        //Convert FinalCount to ASCII, base 10(dec)
                        returnASCII = ltoa(localFinalCount, shortbuf, 10);

//*********************************Diagnostic*****************************
*************
                        if ( TIFR2 & (1<<OCF2A) ) { //*JDW6+JDW3 Counter is reset
every 1ms (1kHz clock rate)

                              TIFR2 |= (1<<OCF2A); //+JDW3 reset the overflow flag

                              SampleCount ++;                    //+JDW3 increment
sample count every 1ms

                        } //end of TIFR2 & (1<<OCF2A)
//*************************************************************************
*************

                        //append FinalCount value to transmit package
                        package = strcat(buffer,returnASCII);
```

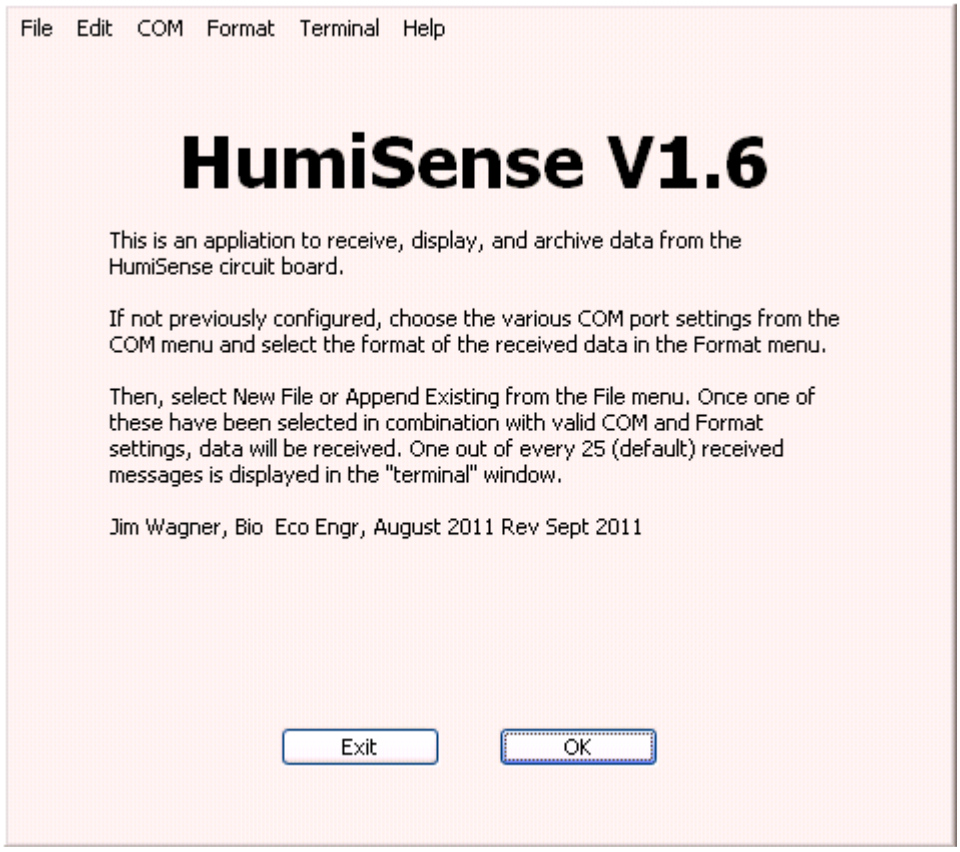
```
                    count_status = kWait;             //+JDW3 computation
done, now wait for 25ms timeout to send

                    //IOPORT &= ~(1<<IO3);      //+JDW5 sets io pin at end of
string construct DIAGNOSTIC
             } //+JDW4 end if count_status = kCompute

             //+JDW3 Test for ADC done. This will really only be used if/when
             //more than one ADC channel is implemented
             if ( ADCSRA & (1<<ADIF) ) { //*JDW6
                    ADCSRA |= (1<<ADIF); //clear the ADC interrupt flag

                    //Storing the ADC conversion value
                    Temp = ADC;

                    //+JDW3 for multi channel, test the current channel
                    //+JDW3 depending on channel, change or leave as is
                    }

             //+JDW3 Test Timer2 for roll-over and deal with state change if
             //elapsed time is 25ms.

             if ( TIFR2 & (1<<OCF2A) ) { //*JDW6+JDW3

                    TIFR2 |= (1<<OCF2A); //+JDW3 reset the overflow flag

                    //cli();
                    //IOPORT |= (1<<IO1);       //+JDW5 sets io pin at start of
transmit DIAGNOSTIC
                    SampleCount ++;                   //+JDW3 increment sample
count every 1ms
                    //IOPORT &= ~(1<<IO1);      //+JDW5 clear io pin at end of
transmit      DIAGNOSTIC
                    //sei();

                    if (SampleCount >=100 ) {   //+JDW3 test for 100ms

                           //IOPORT |= (1<<IO2);       //+JDW5 sets io pin at
start of transmit DIAGNOSTIC

                           //+JDW7 to correct for low frequency error case in
which
                           // there are so few input capture events that the
state
                           //is never properly changed.
                           if (count_status != kWait) {

                                  //see code, above, for documentation
                                  package = ltoa(Temp, buffer, 10);
       //temperature value
                                  package = strcat(buffer,",0,0");   //zero
FinalValue & FinalCount
```

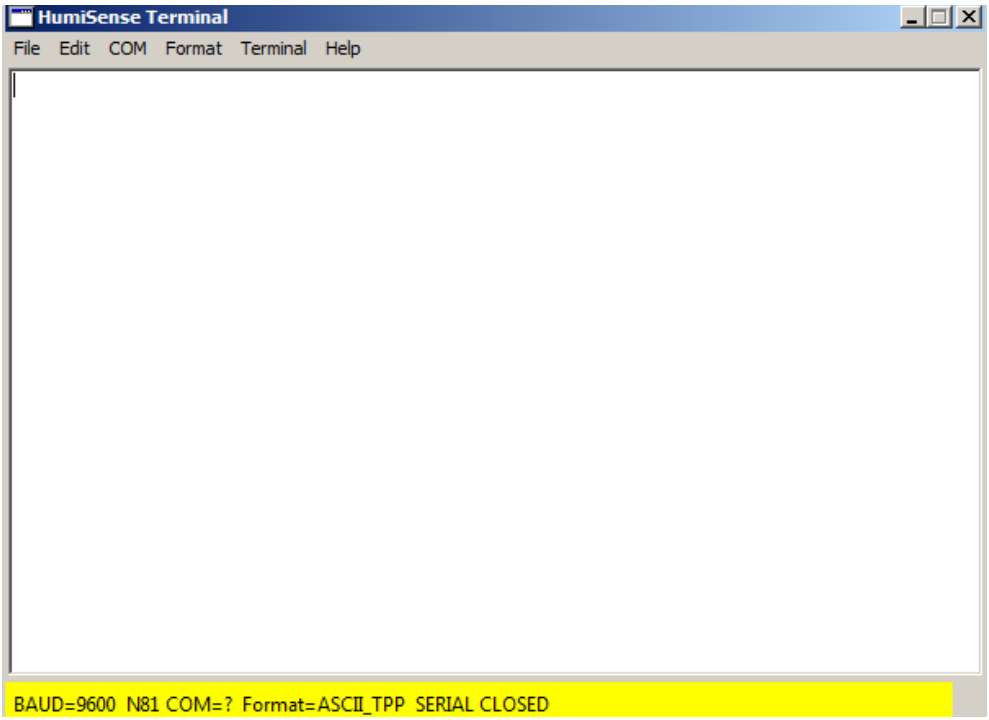
```
                                    avevalue_counts             = 0;
        //for proper operation of ISR
                                    AveValue                    = 0;
                                          //ditto
                                    }     //+JDW7 end

                              SampleCount = 0;            //+JDW3
                              count_status = kMeasure;//+JDW3 start a new
measurement

                              //set the ADC channel - might be different in
multichannel
                              ADCSRA &= kADC_clear;       //clear the channel
selection
                              ADCSRA |= (1<<ADSC)|kADC_Chan3;    //start the
conversion

                              TransmitASCII(buffer);   //*JDW (moved) the whole
string value & CR
                              //IOPORT &= ~(1<<IO2);      //+JDW5 sets io pin at
start of transmit DIAGNOSTIC
                        } //end if SampleCount = 100



              } //end if OCF2A set in TIFR2

       } //end of while loop
} //+JDW4 MOVED FROM ABOVE end of main()
```

Appendix 3. Screenshot of the variable frequency oscillator Excel model



Appendix 4. Screenshot of the application software HumiSense Terminal.




HumiSense Terminal
File Edit COM Format Terminal Help
BAUD=9600 N81 COM=? Format=ASCII_TPP SERIAL CLOSED