Master's degree in Electronic Engineering

Capacitive Sensor Frontend Using Amplitude And Phase Demodulation

Muhammad Rashid

Supervisors
Prof. Mihai Lazarescu
Prof. Luciano Lavagno

Politecnico di Torino
October 14, 2019

Abstract

Indoor human localization sensors are commonly used in smart homes and buildings. These sensors play a vital role and starting point for smart technologies. Imaging, Radio frequency, ultrasound, infrared, pressure and capacitive sensors are generally exploited for this purpose. Every technique has its own special characteristics in terms of cost, sensitivity, accuracy, power consumption and privacy. Another factor is the ease of operation and intrusiveness. So, a sensor will be good if it is passive, sensitive, accurate, low power and unobtrusive.

Capacitive sensing has specific advantages over others. For example, they can be effective in sensing up to range far greater than their size, they observe privacy and unobtrusive. They do not require any wearable device or any special interaction. It is one method which was my task to deal with. It is extensively used in touch screens of laptops, smartphones, tablets etc. to sense touch of human and to measure its movements. I focused on sensors that detects presence as well as location of human body indoors.

Circuit for this frontend was designed by previous researchers. It is based on demodulation of a carrier whose amplitude and phase are affected by presence of human. Using PWM channel of microcontroller ATmega328P, we generate a carrier sine wave of 10kHz which is passed through RC filter formed by sensor plate and tuning resistor. The capacitance gets affected by environmental effects and capacitance variations affect the amplitude and phase of the signal. This affected sine wave is compared to original carrier using a differential amplifier resulting in a new sine wave. The RMS value of this new sine wave is read using an analog to digital converter of the microcontroller for its final wireless transmission. There is signal conditioning i.e. low pass and high pass filters all along wherever required.

My task was to integrate different components, power up, test, debug, characterize the blocks and the system and obtain experimental results, compare and come up with some conclusions based on all previous activities. My approach was to firstly check the schematics of circuit according to theory and datasheets of components. I found some errors in schematics such as wrong connections and some values of components other than specified in datasheets. So, I rectified them in simplest possible way.

Next step I needed to test the circuit for some major faults e.g. to verify voltage of regulators and current of whole circuit are not out of range. One regulator was not working properly, and I replaced it with a new one and placed input capacitor close to it as this was not so before.

As all components were properly powered, I started testing of operation of circuit. First, I split the circuit into different stages and started testing it stage wise. Using multimeter and oscilloscope, keeping theory of research paper in mind, board schematics and layout in front I found the board is behaving correctly. The microcontroller was generating a sinusoid with a low amplitude. I found that it was a software problem. I changed the PWM duty cycles to achieve maximum duty cycle which resulted in a suitable peak to peak voltage. This increased the signal to noise ratio. Another problem was with timing of software code in MATLAB responsible for reception. Previously it was receiving some spurious values from sensor. So, I corrected this too.

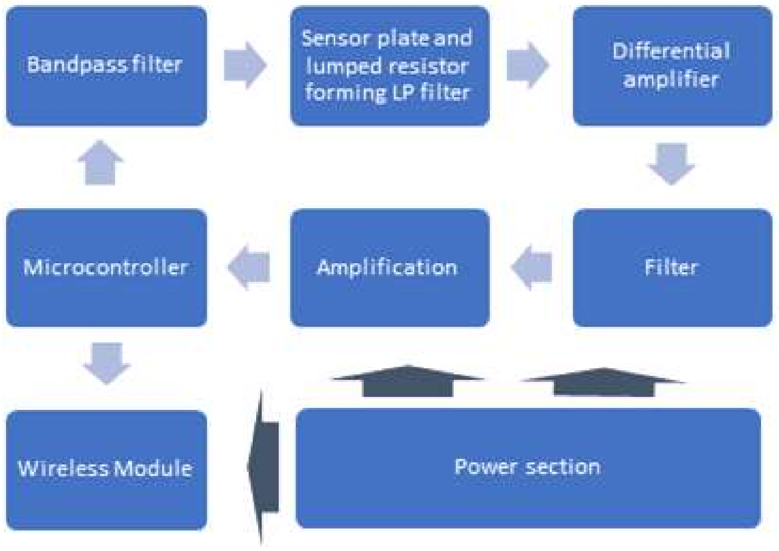

Fig 1: Block diagram showing different stages of sensor frontend

This gave me a level of confidence to move my work to next level. Now I needed to fine tune it according to a good sensor specification i.e. minimize wrong readings due noise and drift, maximize sensitivity, range and stability. As said earlier I used an off-the-shelf wireless transceiver (XBee s1) which made our work quite convenient to receive data on MATLAB for processing and analysis. This stage consumed a lot of time as it was baseline to final operation of sensor. I ran different type of tests i.e. “fast check” used to check whole functionality in quick means. Other test was sensitivity test in which a person stands in front of sensor plate at different distances (predefined) and recorded the readings. Last test was stability test in which the device is kept running for few hours and checked for stability and drift along these hours. This test was performed using a fixed capacitor at first and then using sensor plate of size 16cm×16cm.

By performing all tests in different configurations, I come to know that there are two main problems. One is noise and other is large drift over time. Moreover, I realized that noise in measurements was mainly due to noise present on power lines. So, I tried to reduce it to its minimum level by introducing some appropriate capacitances and replacing some of the components. I also mounted regulator circuitry on a bare board in order to verify its functionality. The problem of drift was due to high quality factor active band pass filters. As I replaced them with passive filters the problem of huge drift was resolved. Another thing I observed that as frontend measurements are based on variation in phase as well as amplitude, so they are sensitive to many external interferences. Therefore, power source, circuit and sensor plate should be kept only in an isolated area having very less disturbance.

Acknowledgment

First of all, I am extremely grateful to God the Almighty for all his blessings. I am very thankful to my supervisors Prof. Mihai Lazarescu and Prof. Luciano Lavagno for their guidance and full support for my work.

I would like to acknowledge Mr. Osama bin Tariq and other colleagues for their help and support in lab work.

Contents

List of Figures

Chapter 1: Introduction

The world of technology is expanding with such a momentum that it is becoming part of our everyday lives. From industry to homes, from high rise buildings to transport, everything is getting smarter. There is a big importance of sensors in smart life. Every decision by a computer heavily depends upon the data it receives from sensors. To make a building or home to be smart human localization is one of the primary objectives.

Thinking with complete engineering point of view it seems to be simple that all people wear some device or garments should have some device with sensors and transmitters that can serve the purpose. But this idea is not applicable in a sense that smart systems are meant to be giving comfort to humans rather than attaching some extra payloads and active transmitters to human body. In this context idea for this type of sensors must be completely passive and not to be worn or carried by persons.

This system should be aware of the privacy of people. No unauthorized person should be able to receive or understand the localization information of people in building. Moreover, it should not require any type of communication or interaction of users with the system. The system must be standalone, analyzing the data and providing relevant results.

Another requirement is about power consumption of the sensor. It should consume very low power that results in long usage on a single battery charge. Low power consumption will also help us reducing battery size resulting in size reduction and weight reduction.

There are different types of sensors having different technologies, each of them having its own specific characteristics.

Ultrasound sensors [3] employ an ultrasound speaker and an ultrasound microphone. Reflection of sound waves can be received and measured in time and frequency with precise calculation of position of person. Moreover, thanks to doppler effect calculation which can also estimate speed of moving objects as well as direction of movement. But this method is costly and being an active system has some other side effects such as it is annoying and harmful for animals.

Human localization can also be done using *radio frequency* [2]. Human bodies absorb RF signals. So, if an RF signal is transmitted, the returned intensity of that signal received at sensor can estimate the location of person. Perfectly calibrated systems can achieve very high accuracy but the major concern about this method is the health of person who is continuously under RF waves.

Another common method for human localization is *pressure sensors* [3]. Sensors are installed in floor of room. They can even identify a person corresponding to his weight. Pressure sensors are installed in a grid structure under the pavement. Accuracy depends on resolution of sensor grid. Main problem and costly process in these sensors is installation and maintenance of sensors.

Image processing is another technique which is completely passive. Captures photo or video and process it using different software techniques. This system is very accurate and can determine not only the position of human but his speed. Moreover, it can have capability of identifying a human being. The drawbacks of this system are costly in terms of computation, it always requires a line of sight between camera and object, there must be appropriate conditions for image/video acquisition. It also raises privacy issues.

Our focus is on capacitive sensing. My goal in this work is to optimize and tune the capacitive sensor in terms of signal to noise ratio, power consumption, stability and maximum achievable range. A lot of researches have been done in this area.

Capacitive sensing

Nowadays capacitive sensing is widely used in touch screens of smartphones, laptops and tablets. The boom expanse of touch screen technology has resulted that we can say capacitive sensing is everywhere. It is a quite easy method of sensing touch and proximity. Human bodies couple with environment in terms of capacitance, therefore sensor can sense the position of human.

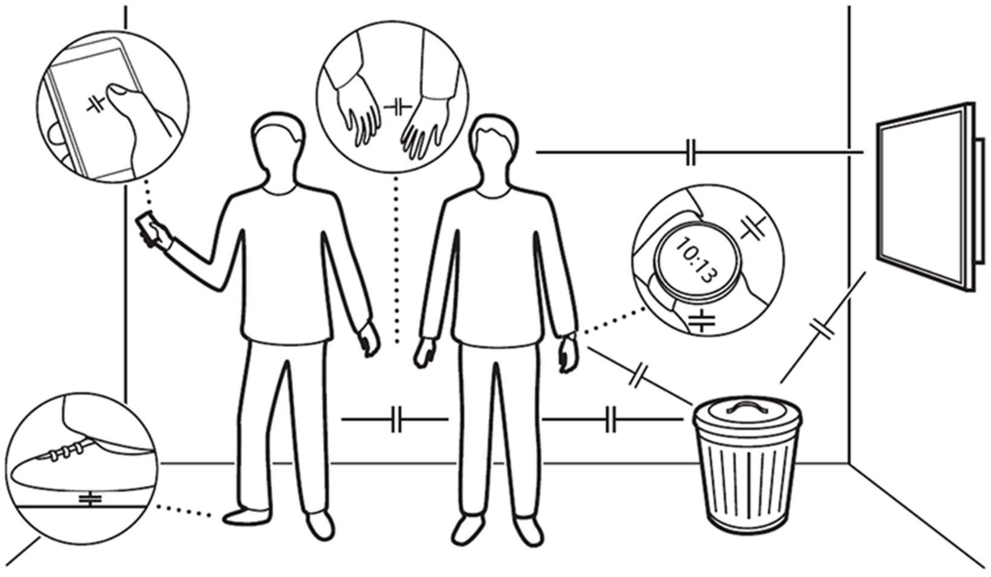

Figure 1.1: Capacitive coupling in a room between different masses from [4].

As Figure 1.1 shows formation of capacitances between the human bodies and other objects. We can consider the human body as a single plate of a parallel plate capacitor and air acts as a dielectric. So, two conductive plates separated by a dielectric develops voltage V. From basic theory of capacitance, we know that

$$\boldsymbol{Q = C.V} \tag{1-a}$$

Where Q is total charge held by plates and C is capacitance. However, capacitance depends on some other factors too. When A>>d capacitance is directly proportional to effective area A of plate and inversely proportional to distance d between them. A constant describing material characteristic also influences capacitance.

$$\boldsymbol{C = \epsilon . \frac{A}{d}} \tag{1-b}$$

Where A>>d. ϵ is permittivity of material. This is not valid for our case because d>>A for long range capacitive sensors. Formula for our sensor can be approximated as follows:

$$\boldsymbol{C \cong k \ . \ \frac{A}{d^{2\sim3}}} \tag{1-c}$$

If we have a look at Figure 1.1 and Figure 1.2 here are four different types of capacitances in this case as follows:

1. C_{sb}: capacitance between human body and sensor plate
2. C_{sg}: capacitance between sensor plate and ground
3. C_{bg}: capacitance between human body and ground
4. C_{se}: capacitance between sensor plate and environment e.g. other significant objects in room

We know from theory that capacitance is a function of distance between two plates and it is also affected by humidity and temperature. In our case C_{sb} is dependent on distance between the human body and sensor plate, which is our interest in measurements.

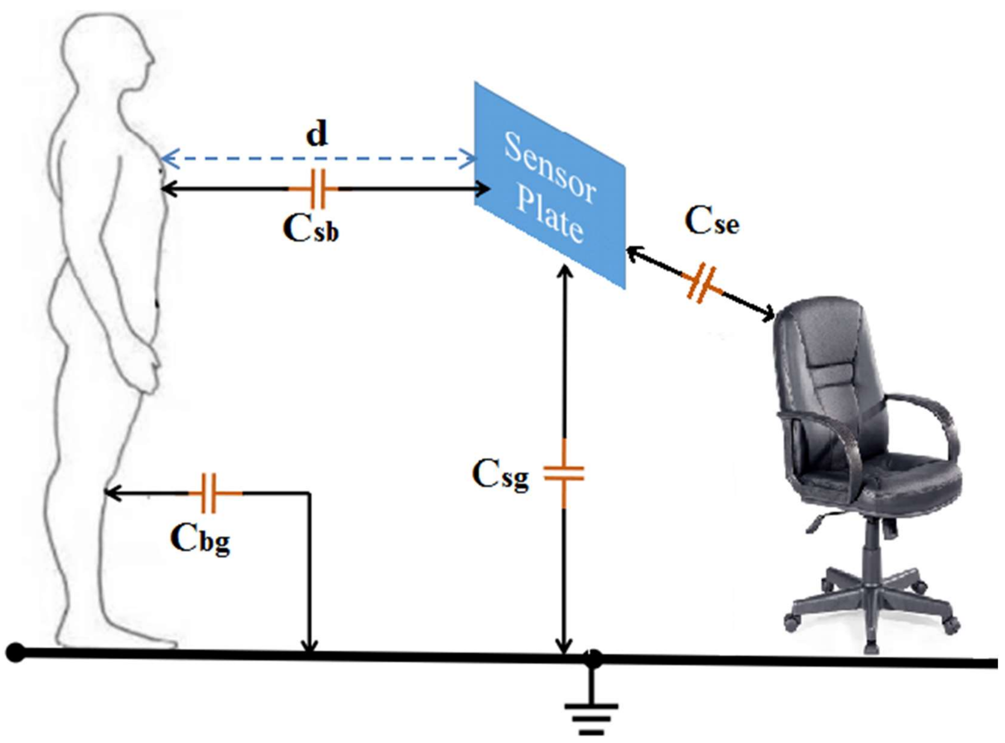

Figure 1.2: Capacitive coupling in a localized area

Working modes

Considering the framework appeared in Figure 1.3 depicted in [5] there are primarily three working modes for the capacitive detecting. Two of them include the utilization of two sensors, one for transceiving.

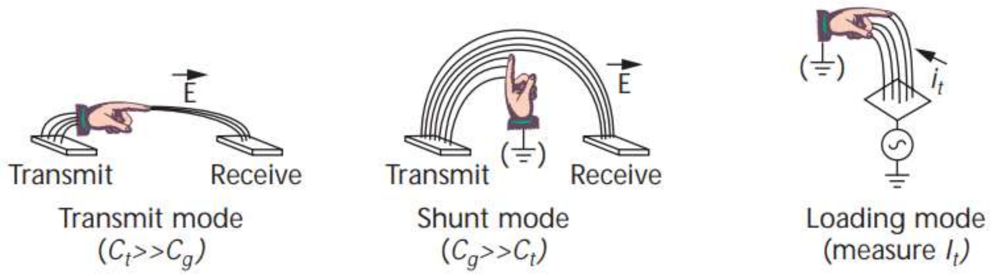

Figure 1.3: Working modes of capacitive sensing

Transmit mode enables the human body to turn out to be a piece of the transmitter, with the increase in received signal relies upon the body closeness. This mode is workable for close distances between the body and the transmitter, because of the significance of the capacitance among body to plate contrasted with the body to ground capacitance.

Shunt mode happens when the body isn't close to the plates and the body ground capacitance wins. For this situation the signal received is diminished with the closeness of the body.

These two strategies are extremely amazing for amount of information. For instance, the transmit mode is truly adept at following developments when the human is making physical contact with the sensor. In any case, alluding to the past discussion, this would not be a simple and tag-less technique.

Loading mode is simple since it needs a single plate. It permits to quantify the induced current in the plate. Both body and plate are alluded to a ground, that is a common potential.

Previous Research

There is a lot of research done on human localization using capacitive sensing using different techniques. Some of them are [6], [7], [8]. The researchers performed different types of experiments in different conditions to obtain maximum amount of data. This data was used to draw some conclusions using knowledge of theory. This project was started by some previous researchers [1] and they designed a simple as well as low cost analogy for this type of frontend.

The circuit was based on measurement of amplitude and phase difference which was caused by the change in capacitance. Copper plate of 16cm×16cm was used as a sensor mounted on the wall of a room. The room was about 4m×4m.Sensitivity of 200cm was obtained with good resolution and low noise. Using which human was accurately localized in the room by attaching a sensor on each wall.

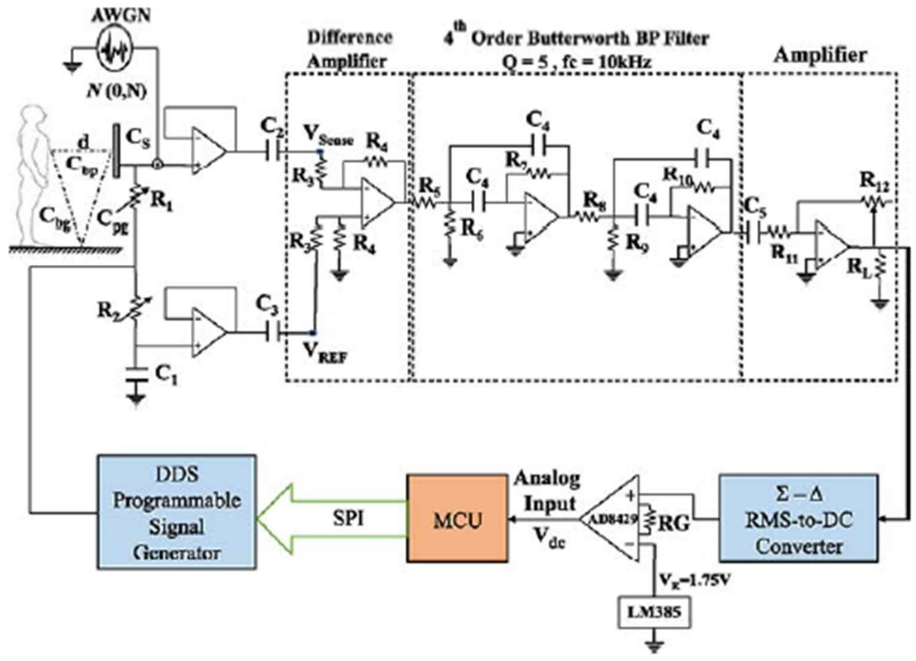

Figure 1.4: Circuit design of frontend interface (from [1])

This circuit shown in Figure 1.4 detects and measures the difference in capacitance caused by human coupling. Sine wave of 10kHz is generated on board using DDS programmable signal generator and fed to both branches having RC low pass filters. However, on one branch sensor plate is used as a capacitor shown as C_s. The sinusoidal signal passing through RC filter formed by sensor plate and a variable resistor is affected in terms of phase and amplitude. DC offsets are also removed using C_2 and C_3 and then compare both using a difference amplifier. The resulting signal is also a sinusoid which is passed through high Q-factor band-pass filters to remove noise. A sigma-delta RMS to DC converter was used to convert this sinusoid to a DC value. This DC voltage is measured by an internal analog to digital converter ADC of ATmega328P microcontroller by Atmel corporation. Microcontroller is also responsible for controlling signal generator using SPI interface.

Mathematical model explained in [1] explains that:

$$\mathbf{V_{in} = A\angle 0^o}$$

And

$$\mathbf{V_{Ref} = A_2\angle\phi_2 = A_2 e^{j\phi_2} = A_2\left(\cos\phi_2 + j\sin\phi_2\right)}$$

Environmental noise entering from sensor plate is modeled as zero-mean additive white Gaussian noise (AWGN), So:

$$\mathbf{V_{Sense} = A_1\angle\phi_1 + \mathcal{N}(0,N) = A_1 e^{j\phi_1} + \mathcal{N}(0,N)}$$
$$\mathbf{V_{Sense} = A_1\left(\cos\phi_1 + j\sin\phi_1\right) + \mathcal{N}(0,N)}$$

We can observe that gain of differential amplifier is R4/R3. Therefore, output will be:

$$\mathbf{V_{Diff} = \frac{R_4}{R_3}\left[V_{Ref} - V_{Sense}\right] + \mathcal{N}(0,N)}$$

$$\mathbf{= \frac{R_4}{R_3}\left[\left(A_2\cos\phi_2 - A_1\cos\phi_1\right) - j\left(A_2\sin\phi_2 + A_1\sin\phi_1\right)\right] + \mathcal{N}(0,N)}$$

Using 4th order Butterworth bandpass filter with a center frequency of 10kHz the additive noise is reduced, and the result can be approximated to

$$\mathrm{V_{out}} = \frac{\mathrm{R_4}}{\mathrm{R_3}}\left(\mathrm{A_2}\cos\phi_2 - \mathrm{A_1}\cos\phi_1\right)$$

$$-\mathrm{j}\frac{\mathrm{R_4}}{\mathrm{R_3}}\left(\mathrm{A_2}\sin\phi_2 + \mathrm{A_1}\sin\phi_1\right)$$

with

$$|\mathrm{V_{out}}| = \mathrm{G}\sqrt{\mathrm{A_1^2} + \mathrm{A_2^2} + 2\mathrm{A_1}\mathrm{A_2}\cos\left(\phi_2 - \phi_1\right)}$$

where:

$$\mathrm{G} = \frac{\mathrm{R_4}}{\mathrm{R_3}}$$

$$\mathrm{A_1} = \frac{1}{\sqrt{1 + \left(\omega \mathrm{C_S} \mathrm{R_1}\right)^2}}$$

$$\mathrm{A_2} = \frac{1}{\sqrt{1 + \left(\omega \mathrm{C_1} \mathrm{R_2}\right)^2}}$$

$$\phi_1 = -\arctan\left(\omega \mathrm{C_S} \mathrm{R_1}\right)$$

$$\phi_2 = -\arctan\left(\omega \mathrm{C_1} \mathrm{R_2}\right)$$

In above equations to last all parameters are known constants so output voltage is function of C_s.

$$|\mathrm{V_{out}}| = \mathrm{f}(\mathrm{C_S}).$$

As discussed previously that C_s is also function of many other factors such as humidity, temperature, position and movements of other objects in the room.

For highest sensitivity of the sensor, R_1 and R_2 were adjusted in such a way that

$$\mathrm{A_1} = \mathrm{A_2} = \frac{1}{\sqrt{2}}$$

$$\phi_1 = \phi_2 = -\frac{\pi}{4}$$

The above condition makes both branches balanced which result in non-monotonic dependency of output of sensor due to C_s variations. So R_2C_1 is little detuned in such a way output from R_1C_S is never going to match output of other.

Chapter 2: The Project

Description

This project started considering previous research [1] as a baseline discussed in chapter 1. A developed PCB assembly was available for purpose of testing and improvements. Software code for analog value reception using in-built ADC and transmit this data using UART interface was available. This UART was interfaced with an off-the-shelf wireless module Xbee S1 that transmitted the reading to the PC. On PC end there was a same XBee S1 module configured for reception at specified baud rate (9600 bits per second) connected through USB port. I used MATLAB for data reception and recording purposes as it was quite convenient to use as well as to analyze data.

The frontend electronics was composed of six rail-to-rail input and output operational amplifiers (OP184) in 8-lead SOIC package. Two of them were used in a band pass Sallen-Key filters while one of these band pass filters was responsible for construction of sinusoid of frequency 10kHz. Another two of them were input followers in unity gain configuration. The purpose of them to use was to have a good isolation in terms of impedance as we know that operational amplifiers have a large input impedance and very small output impedance. One was used in differential mode while the last one was just an amplifier with adjustable gain.

There's also an instrumentational amplifier (AD8429) used after the band-pass filter. The purpose of using specifically this type of operational amplifier is that it is designed to deal with small signals from sensors and has very good common mode rejection. In a normal differential amplifier, we have combination of resistors to perform this operation and these resistances are never completely matched. Therefore, CMR of a common operational amplifier can't be larger or equivalent to that of an instrumentational amplifier as it uses a single resistor to adjust the gain. Three voltage regulators for 5V, 10V (±5V) and 3.3V are used to provide appropriate power to the components. Copper plate of 16cm×16cm used as a sensor was fixed on a dummy wall on chest height (estimated).

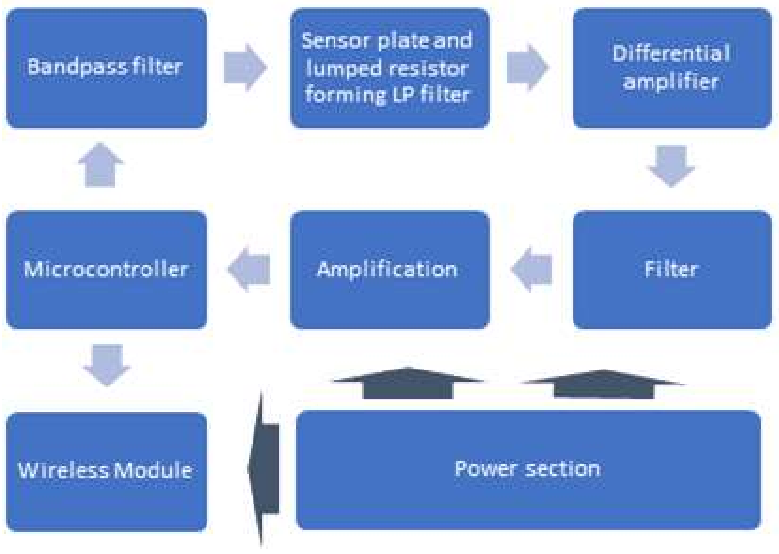

Figure 2.1: Block diagram of sensor frontend

Operation

Figure 2.1 shows design flow of all blocks. Starting from the microcontroller, it generates 10kHz frequency PWM signal which is passed through a filter resulting in a clean sinusoid of same frequency. This sinusoid is split into two branches with RC filters, one with lumped capacitor with an adjustable resistor while other with a sensor plate and an adjustable resistor. As discussed in chapter 1 that human bodies couple with environment in terms of capacitance. So, capacitance of plate is dependent on presence of human body or any other object in front of it considering a basic capacitor. Equation (1-c) tells us that capacitance is inversely proportional to the distance between the plates of capacitor. As a result, the sinusoid signal on the branch with sensor plate as capacitor is affected in terms of amplitude and phase.

This change is measured using differential amplifier and feed it to an RMS-to-DC converter that converts this sinusoid to its corresponding DC value. This DC level is eventually measured at ADC after passing it through some amplifications and filters. In this application, we are interested in movements of humans (slow movements) so we can easily neglect any sudden change in amplitude. That's why we use filters here.

Analog to digital converter ADC reads the value and sends to XBee module using its UART interface which is then transmitted on specified baud rate.

Work Methodology

Now I will share different steps involved in my research work. These steps are not exactly consecutive in time. But some activities of these steps were simultaneous.

Schematics Verification

I started my work from the schematics and the datasheets of used components. Firstly, I needed to understand role of every component and section. So, I consulted datasheets of components for their characteristics and transfer functions. Soon I realized that there was a wrong connection as shown in Figure 2.2

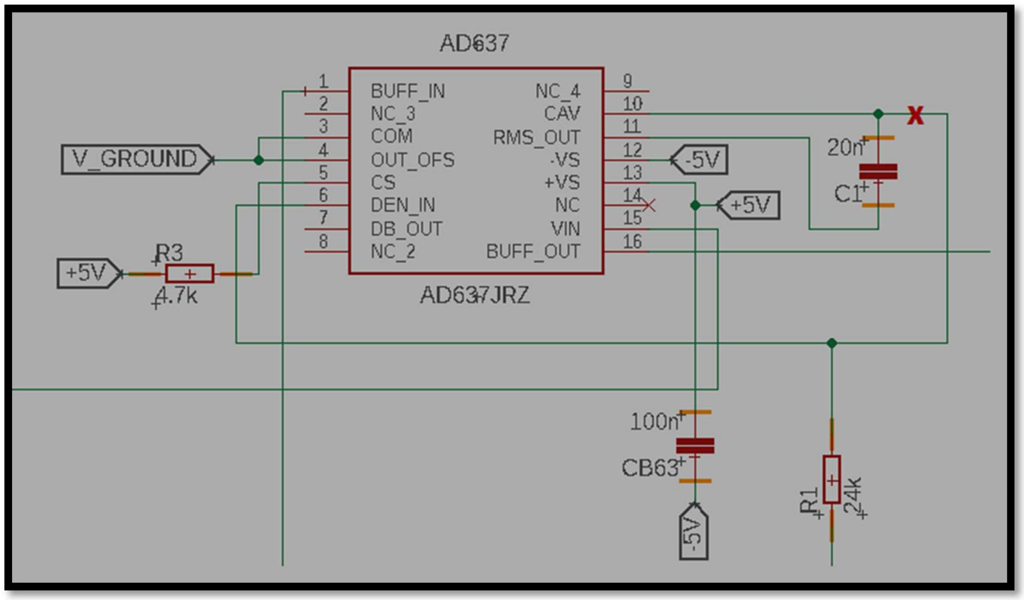

Figure 2.2: Wrong connection in schematics

AD637JRZ is an RMS to DC converter IC. It converts AC sinusoidal signal to a DC value equal to its root mean square value. Datasheet of AD637 suggests that ripple reduction method is preferable to use with this component. So, it provides a two-pole low pass Sallen-Key filter shown in Figure 2.3.

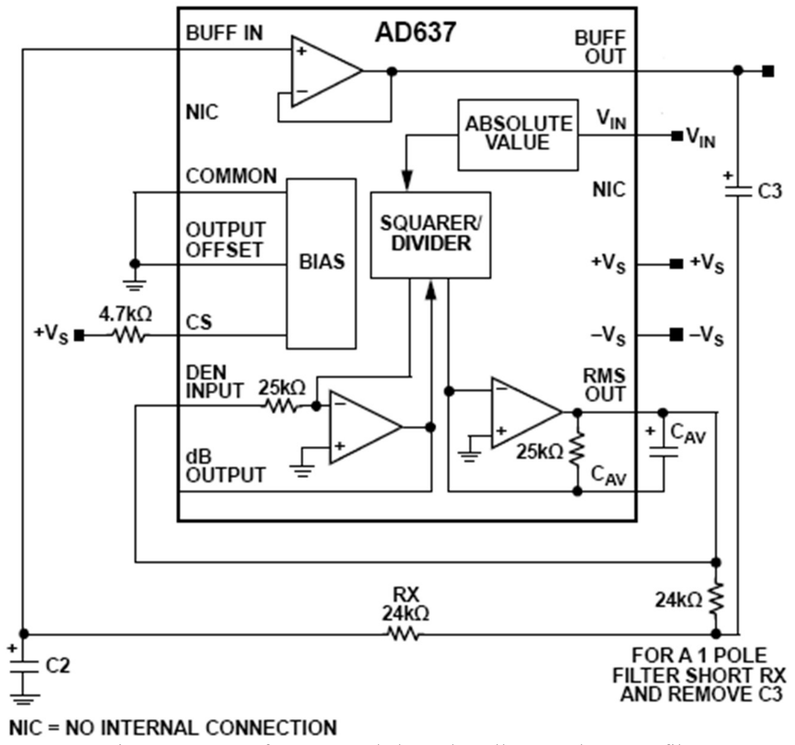

Figure 2.3: Use of recommended 2-pole Sallen-Key low pass filter

Here we can observe that two pins named DEN INPUT and RMS OUT are connected to each other while in schematics DEN_IN is connected to CAV which is not correct. So, I must connect it according to the datasheet. Firstly, it seemed very simple but when I studied the PCB layout and the circuit physically, I realized that it is not as simple as it seems to be.

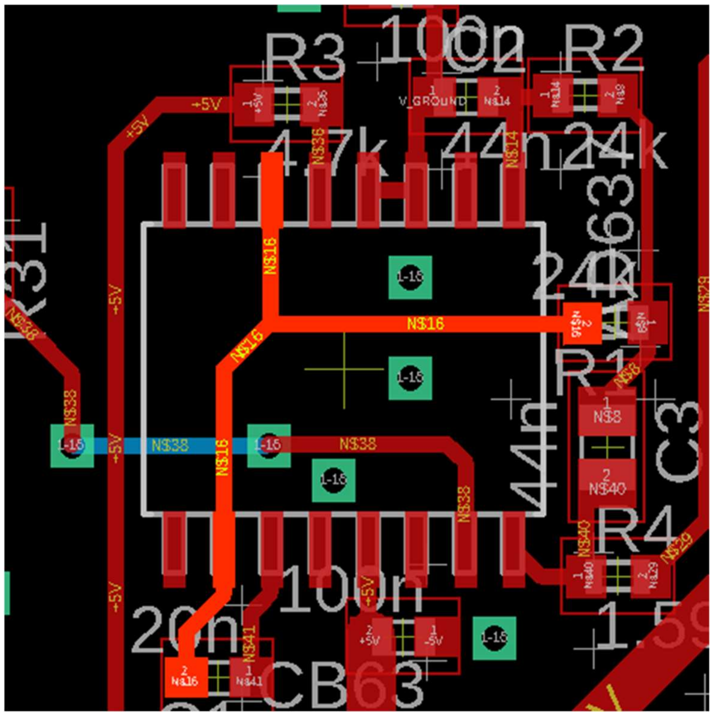

Figure 2.4: PCB layout of AD637

As it is evident from Figure 2.4 that the track connecting is passing underneath the IC. One method was to remove the IC cut the track and connect it externally using a jumper wire, but I didn't go for it. I chose the second one which was to keep the IC on its place, lift the concerned pin of the IC and make connection externally using a jumper wire. So, I lifted the pin, cut the external track to capacitor and soldered jumper wire over the components. After this I verified electrical connections using a multimeter.

Moreover, there were few wrong values for passive components which was not so difficult to manage. I replaced those components with new correct values. Every re-soldering process was followed by continuity check using multimeter. So that if there is any problem with soldering process that can get highlighted at earliest.

Verification of Power Supplies

As verification of schematics completed next task was to ensure supply of adequate voltage and to check for overcurrent. Following steps were taken into consideration during this step:

- To check for over-current by measuring overall current of circuit.
- Measurement of different power supplies' voltages and to verify if they are in range or not
- To check noise on voltage supplies i.e. V_{pk-pk} of voltage ripples
- In case of any problem check off-load voltage and on-load voltage.

I powered up the circuit board using a power supply having both voltage and current display using 5V and 400mA setting. Current display didn't show any current limiting and voltage didn't drop. This showed that there is no overcurrent or any short circuit on supply lines.

There were four different supplies generated on the board -5V, V_DIG (0V), +5V and 3.3V. -5V and 0V is used for digital section i.e. microcontroller which runs on 5V supply. -5V and +5V are used for analog circuit to deal with bipolar signals. Two 5-pin ICs NCP1403 was used to generate above supplies. A rail splitter TLE2426 was used to generate a virtual ground exactly in middle of -5V and +5V to help in dealing with analog bipolar signals. Supply of 3.3V was generated using ADP2503 only for Wireless module XBee S1 as it requires supply voltage between 2.8V and 3.4V.

When I started checking voltages of these supplies, they seemed pretty good except for 3.3V. for which I realized that the regulator for 3.3V ADP2503 is either not correctly soldered or it is malfunctioning. This regulator was 3mm×3mm QFN package having all the pins underneath. As this component was small enough and very thin pins underneath, it was a difficult task to hand solder it using a heat gun. But consulting some experts and temperature control guide online I was able to do align and solder it successfully. I checked its output voltage it was nearly 3.3V and I was happy with this. But soon I noticed that XBee module is skipping some samples in its transmission to the PC. It took a little time to figure out this problem and the problem was again with 3.3V supply. Off-load supply was clean 3.3V but when I loaded it with XBee module I realized that it drops voltage below 2V that switches off the wireless module (Figure 2.5)

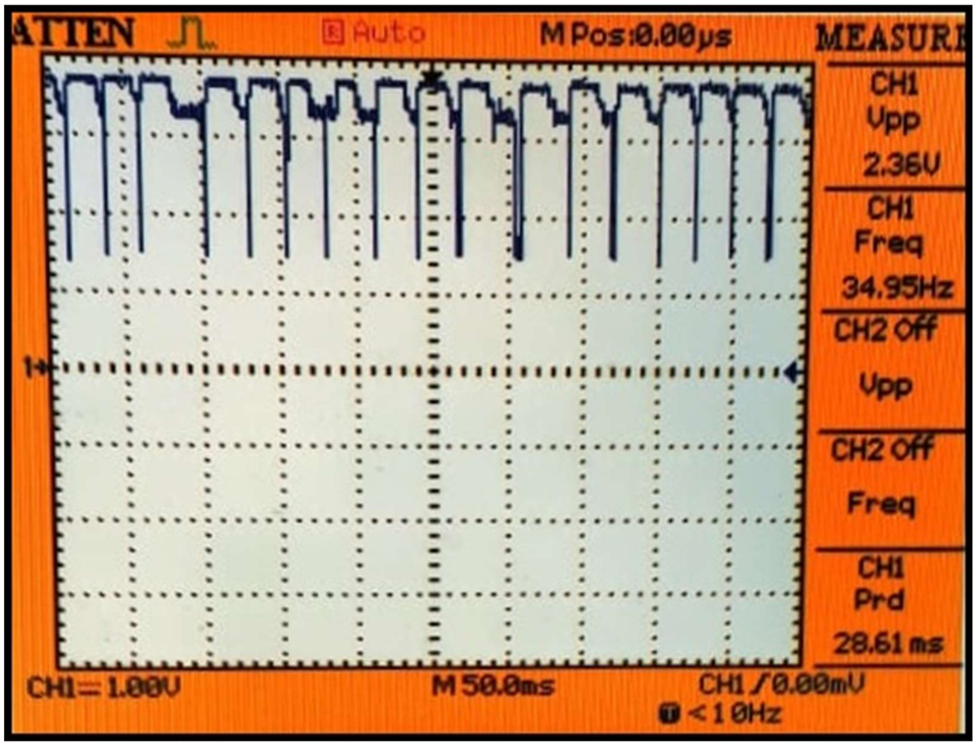

Figure 2.5: Output of 3.3V regulator with wireless module as a load

I started to analyze the PCB layout related to this 3.3V regulator. I realized that datasheet recommends to place input and output capacitor very close to the regulator. But here input capacitor was away from it as shown in Figure 2.6 and this input capacitor was also shared by two other regulators too.

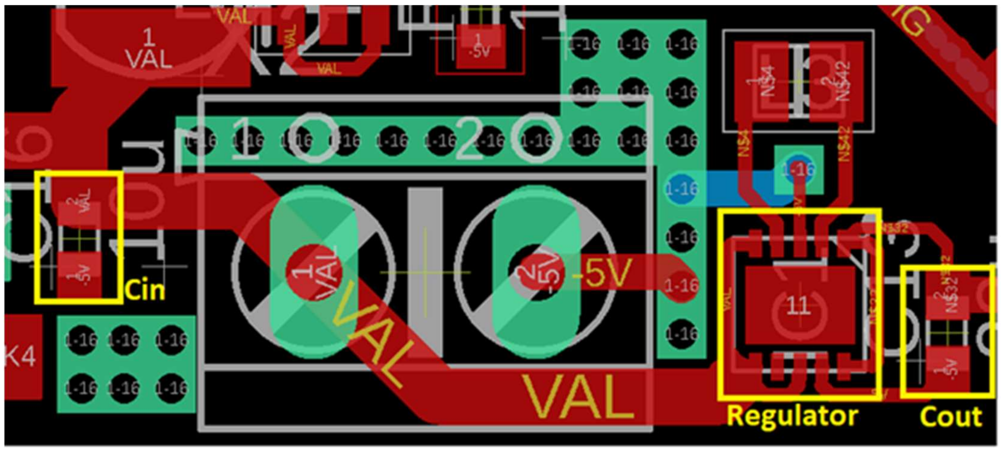

Figure 2.6: Distance of input capacitor from ADP2503 regulator is large

So, it was quite crucial to provide a separate input capacitor to this regulator which must be very close to it. Therefore, I soldered a 10uF capacitor of case 0603 very close to the regulator which solved the problem of onload voltage drop and resulted in correct wireless transmission.

When it comes to noise, I was least concerned about noise on 3.3V supply. Only the voltage should be in recommended range of XBee module. I was mostly concerned about noise on digital supply and analog supply i.e. -5V, 0V and +5V lines. When I checked noise on digital supply V_DIG it was quite high as shown in Figure 2.8

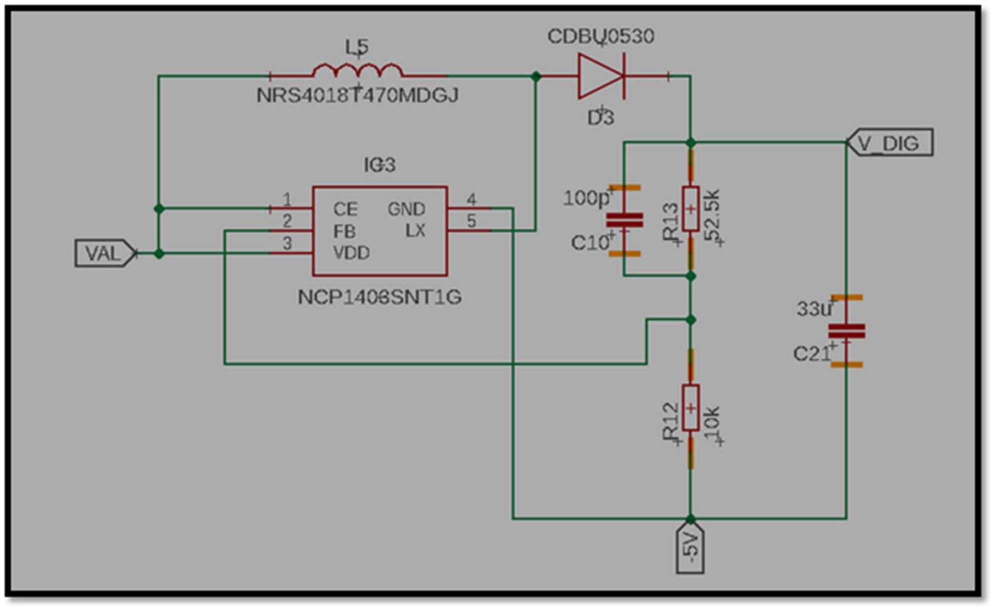

Figure 2.7: Digital power supply (0V, -5V) generated using external 5V

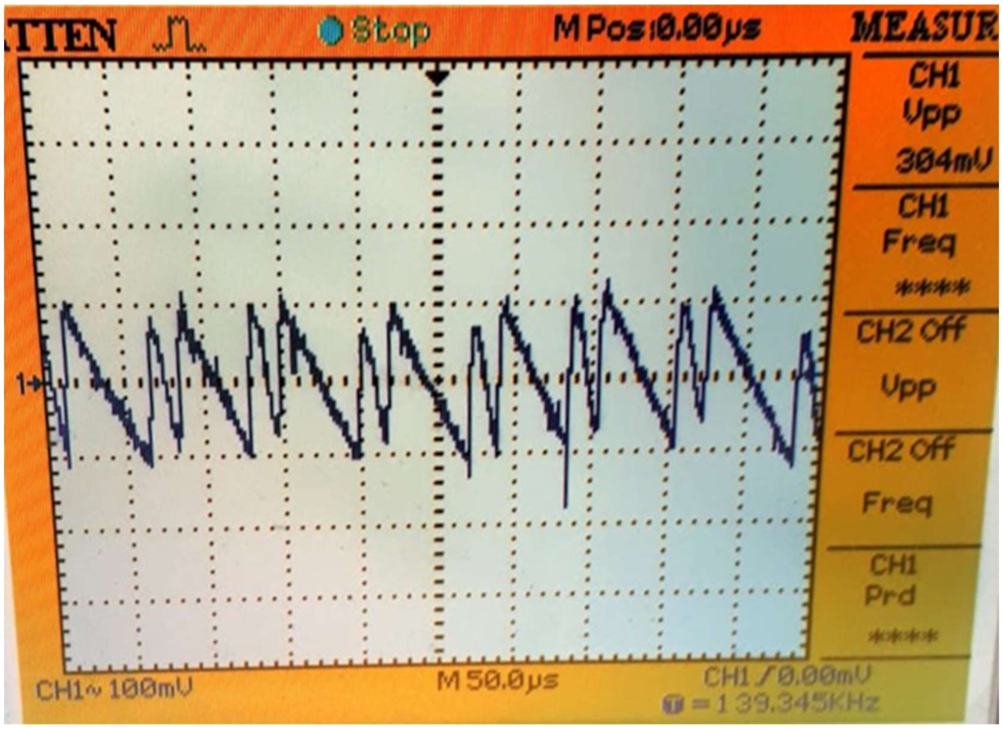

Figure 2.8: Very high noise on digital supply

This amount of noise is totally unacceptable as it is even more than 5%. So, I started to test the regulator separately. I requested a new bare printed circuit board and started to solder a new regulator and its necessary components to have an independent test. Meanwhile I used another inductor with the same value to check if there is some problem caused by inductor. I tested the regulator from low to high current by changing load resistance. I found quite promising performance of regulator even at high currents of 41mA as shown in Figure 2.9.

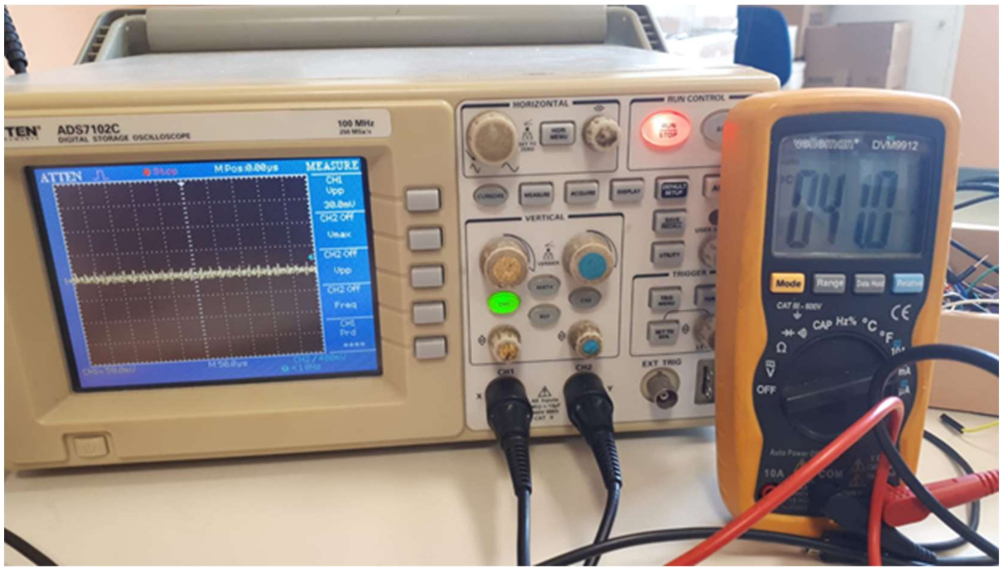

Figure 2.9: Independent load test of regulator on a bare PCB. Digital oscilloscope shows voltage ripples while multimeter shows the current in milli-amperes.

As it is visible peak to peak noise on regulator output is nearly 30mV at 41mA, which means noise level is less than 1%. This gave me a clue that there is problem with inductor or soldering of components. I replaced inductor with new one and allowed the solder to reflow using a heat gun in concerned section of PCB. When I checked again on working PCB it was nearly 50mV (1%) that is quite better. Also, I realized that some of unwanted voltage fluctuations were also reduced.

Functional Run

As whole circuit was powered up properly, it was time to test its functionality. My approach was first to perform a generic run and verify whole functionality of circuit using digital multimeter, digital oscilloscope and wireless reception on MATLAB. Soon I realized the circuit is integrated in correct manner and values of voltage is received on PC and this value also changes with any interruption or touch to any component on board. This gave me a confidence that my circuit is behaving in a correct manner. I started checking the circuit stagewise to understand the behavior of each block (Figure 2.1). For example, I started from 10kHz sinusoid which is generated on board using PWM channel of microcontroller. checked its frequency and amplitude. Frequency was found correct, but the amplitude was seemed to be low. When I checked code of microcontroller it was not using the whole duty cycle range (only half of range was used). So, I calculated new values for 10kHz using maximum range of PWM duty cycle, inserted in code and reprogrammed the microcontroller. This allowed a better SNR and ease in amplification.

During this test I also recognized that when any test is started, some unwanted values are received at the start of every test. This problem was related to timing during test and undesirable values were previous values stored in FIFO of transmitter. I checked the input and output of each block and matched it with theory of operation explained in datasheets (e.g. gain adjustments). Few trimmers were there to be adjusted for appropriate value of gains. I checked transfer functions of operational amplifiers and adjusted gains so that input of ADC on microcontroller driven by the previous analog circuitry is within range of ADC under all conditions. I had a MATLAB test code named "fast check" whose test lasted only for 30 seconds in order to check functionality of circuit and transmission. This task didn't consume much time as output of analog circuit was easily visible on oscilloscope as well as on PC.

Fine Tuning

After generic functional run my next plan was to fine tune the circuit. I started from tuning the cut-off frequency of R_1C_s (R_1 is TRIMM2 in schematics) according to [1] which states that amplitude of output of R_1C_s should be 70.7% of original sinusoid when nobody is in front of sensor plate.

In the same way I tuned cut-off frequency of R_2C_1 and both branches got balanced. I slightly detuned it in a direction keeping the output monotonic and within readable range of ADC. Then I performed some measurement runs to check sensitivity and stability over time.

Problems

I found two main problems visible in my test results, measurements were not stable over long period of time i.e. high drift and there was significant noise. To check stability over time the frontend was kept running for few hours with no human movement (usually at night) in a room of 3m×3m. All ADC values were received and saved in PC memory in array format readable by MATLAB. Figure 2.10 shows a result of seven hour stability check. There was a huge drift of nearly 40mV over 5hours i.e. 8mV/hour (40mV/3.995V = 1%) which was unacceptable.

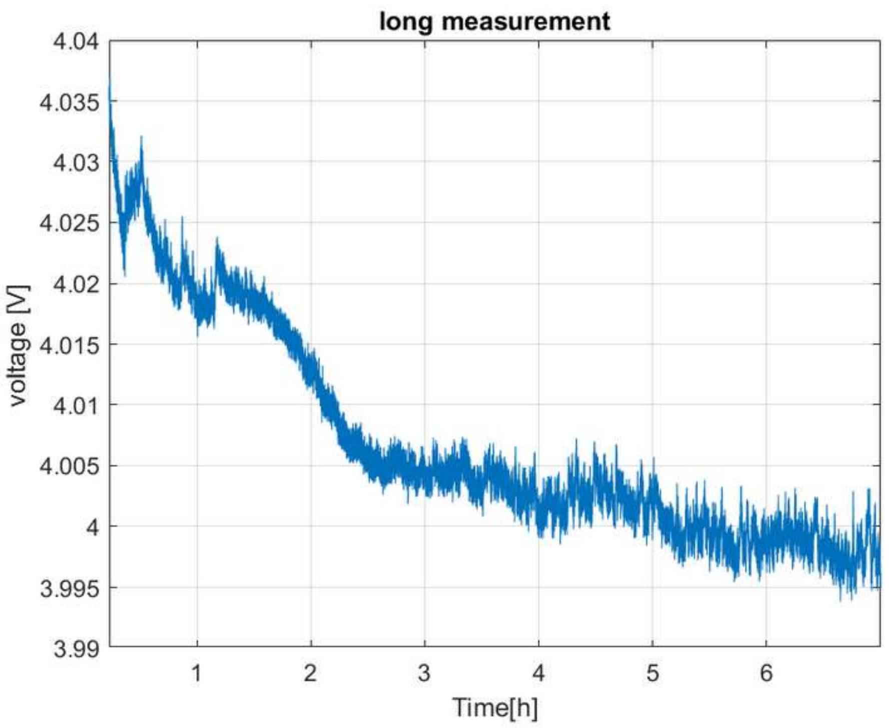

Figure 2.10: Result of stability test high drift over seven hours

To check sensitivity of the sensor a person stood in front of sensing plate on some pre-defined distances and readings were recorded. Figure 2.11 shows a typical example of sensitivity test. Sensor plate is placed at a specific place from which distance is marked on the floor. The person moves to predefined distances according to test scenario. This experiment indicated sensitivity of sensor i.e. amount of change in voltage due to change in distance of person. Moreover, this experiment also revealed the amount of unwanted noise i.e. stability of signal over small period when other factors (especially distance) are kept unchanged. Figure 2.12 shows noise of more than 3mV (3mV/4.059 = 0.073%) which I tried to reduce later.

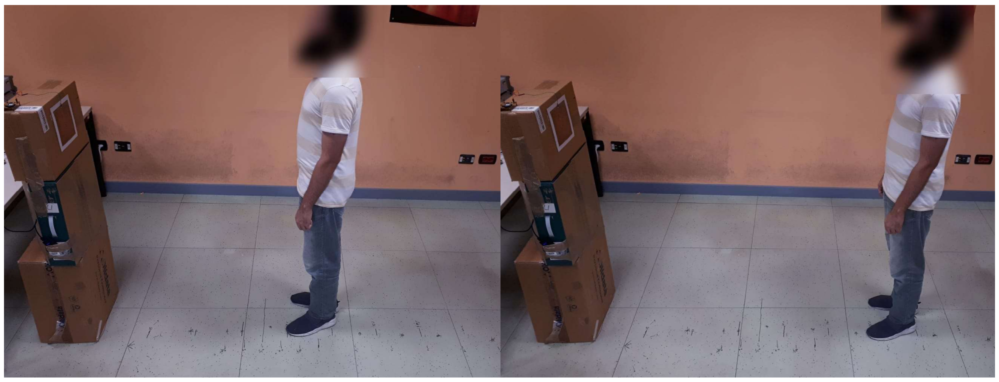

Figure 2.11: Photograph showing sensitivity test

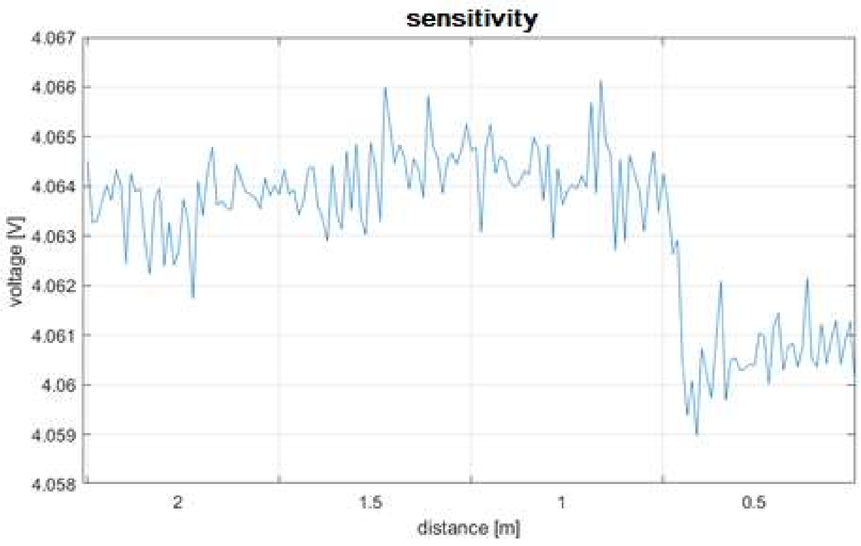

Figure 2.12: Result of sensitivity test for person at different distances (showing high noise)

Solutions

The problem of drift can be because of environmental changes i.e. temperature and humidity as air conditioning of the room was centrally controlled that resulted in change in climate for the sensor. But this effect should be very minimal. I found that the high quality active band-pass active filters caused this drift (Figure 2.13). When I bypassed one BP filter and replaced other with two cascaded passive filters, drift problem was reduced to a negligible level (results discussed in Chapter 3:).

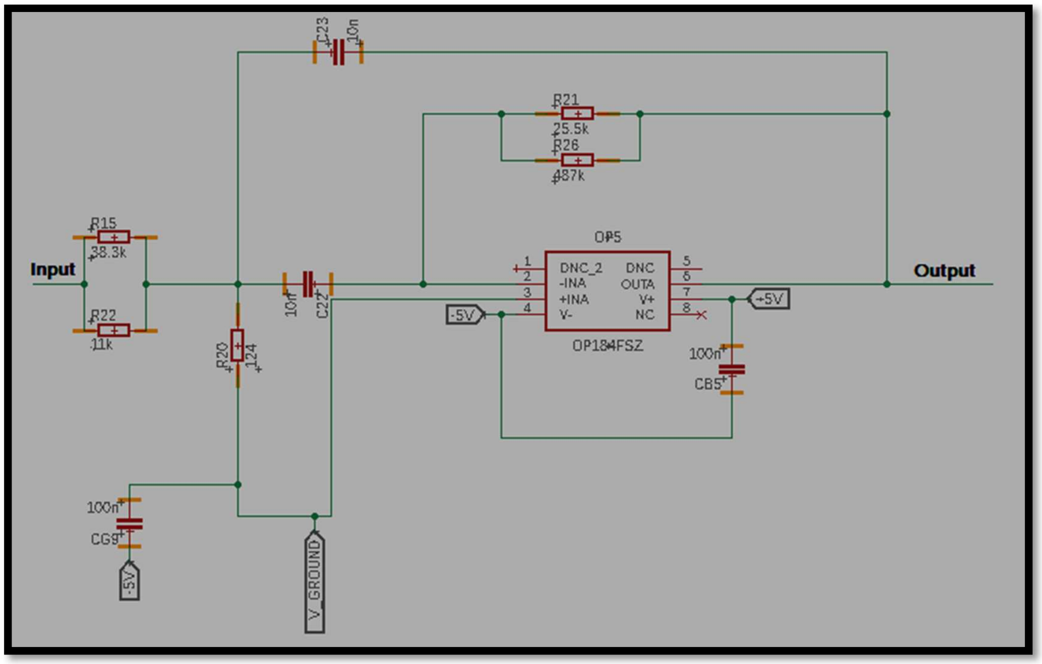

Figure 2.13: High quality band-pass active filter

The problem of noise could not be eliminated completely but with few changes in circuit and some precautions it was reduced to much extent. As this circuit used amplitude demodulation so it is highly susceptible to any type of disturbance whether physical touch, static charge or electromagnetic noise in vicinity. I introduced two low-pass filters of frequencies as low as 10Hz after RMS to DC converter IC that resulted in much less noise. I did some more steps to reduce noise in circuit. I reduced the length of wire connecting sensor plate to circuit to less than 15cm. I replaced the trimmer (TRIMM2) attached to sensor plate with a fix resistor of 2MΩ as trimmers are noisier. I placed the 5V external battery used as a primary source of power far from the circuit. Power cable and cable connecting sensor plate to circuit were kept hanging in air, not touching walls, any other equipment or furniture as they can have static charge. Circuit board was mounted with screws in such a way that body of PCB should not touch anything directly. And last but not least all changes, corrections, introduction of new components were properly soldered and prototyping using breadboard was not used at all. Results are shown and discussed in chapter 3.

Further tuning

Doing all these experiments I considered another option too. As discussed, the difference was measured between output of R_1C_S (branch with sensor plate as a capacitor) and R_2C_1 (branch with lumped capacitor) as a reference. I understood that our reference is already dependent on a low pass filter which is not good. I proposed that difference between output of R_1C_S and original sinusoidal signal should be measured. The reason was to have a fix and stable reference to compare and measure the difference. So, I made some calculations for it.

Let us consider:

$$\boldsymbol{k} = \frac{\boldsymbol{f_c}}{\boldsymbol{f_m}} \qquad \text{(2-a)}$$

where $\boldsymbol{f_c}$ is cutoff frequency and $\boldsymbol{f_m}$ is modulation frequency. Angular frequency $\boldsymbol{\omega_m}$ is

$$\boldsymbol{\omega_m = 2\pi f_m} \qquad \text{(2-b)}$$

$$\boldsymbol{RC} = \frac{\mathbf{1}}{\boldsymbol{2\pi f_c}} \qquad \text{(2)}$$

Let $\boldsymbol{A}$ be amplitude of new signal

$$\boldsymbol{A} = \frac{\mathbf{1}}{\sqrt{\mathbf{1} + \frac{\mathbf{1}}{\boldsymbol{k^2}}}} \qquad \text{(2)}$$

$$\boldsymbol{\varphi} = -\boldsymbol{atan}\left(\frac{\mathbf{1}}{\boldsymbol{k}}\right) \qquad \text{(22-c)}$$

So, the difference signal $\boldsymbol{O}$ will be

$$\boldsymbol{O} = \boldsymbol{sin}(\,\boldsymbol{\omega_m t}) - \boldsymbol{A}\,\mathbf{sin}\,(\,\boldsymbol{\omega_m t + \varphi}) \qquad \text{(2)}$$

For simplicity assume $\boldsymbol{f_m = 1}$ & $\boldsymbol{k}$ is positive

$$\boldsymbol{O} = \boldsymbol{sin}(\boldsymbol{2\pi f_m t}) - \frac{\mathbf{sin}\,(\boldsymbol{2\pi\, f_m t - atan}(\frac{\mathbf{1}}{\boldsymbol{k}}))}{\sqrt{\mathbf{1} + \frac{\mathbf{1}}{\boldsymbol{k^2}}}} \qquad \text{(22-d)}$$

Solving for second integral and plotting it using software wxMaxima 17.10.1 gives

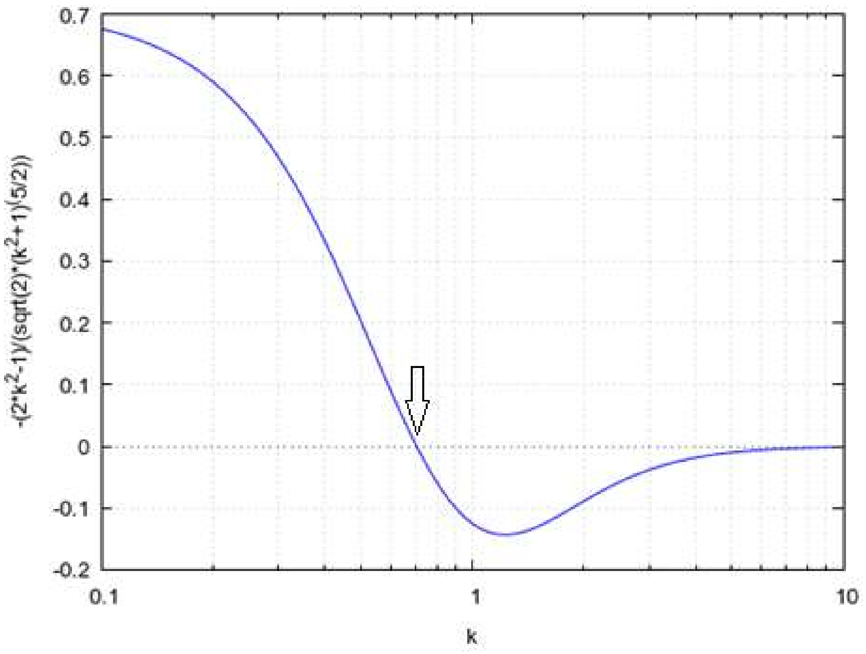

Figure 2.14: Second integral plot for calculation of maximum sensitivity

Calculating amplitude of signal for maximum sensitivity i.e. zero crossing of graph in Figure 2.14 gives us:

$$\boldsymbol{A = \frac{1}{\sqrt{3}}}$$

Which is equal to 57% of the amplitude of original signal. So, I tuned my R_1C_S by changing R_1 to value (2MΩ) which gave nearest value of amplitude of signal.

By doing all changes mentioned above the results were very promising in terms of noise, drift and sensitivity discussed in next chapter.

Chapter 3: Experimental Results

Experimental setup

Talking about experiment setup and conditions are very important before we proceed to the results. The plate with size of 16cm×16cm was mounted on a dummy wall made of cardboard at height of human chest and the circuit was fixed on top of cardboard box little away from plate. This helped in reducing noise on plate, caused by circuit. 15cm of cable was used to connect the plate to the sensor. A power bank by Trip Lite with model number UPB-05K2-1U was used as primary power source. This power bank was at a lower height nearly 60cm below the circuit. Power cable of length 100cm was used. The power cable and the cable connected to plate were straightened in such a way that they should touch only at point of electrical connection as shown in Figure 3.1.

XBee S1 802.15.4 RF modules by Digi International were used at both transmitting and receiving ends. MATLAB R2018b was used for data reception on PC. Digital oscilloscope model number ADS7102C by Atten Instruments and Digital multimeter DVM9912 by Velleman was used when required. All experiments were performed during summer and stability run were done during night.

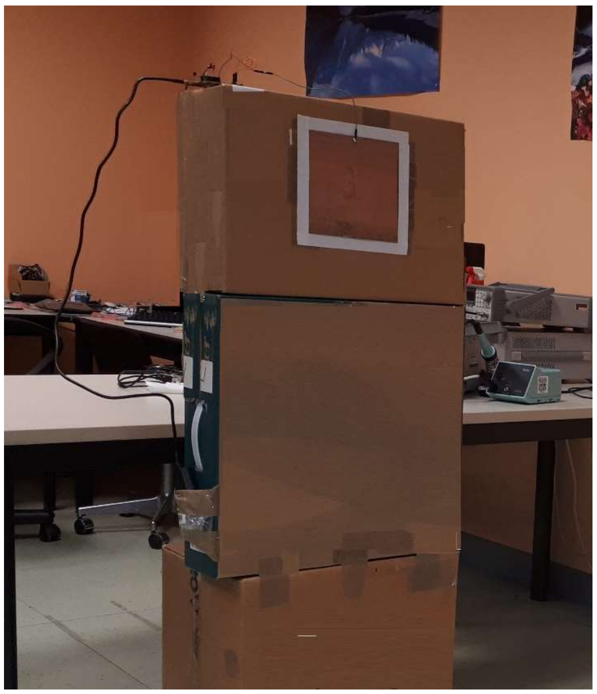
Figure 3.1: Experimental setup

Sensitivity

Problems of noise during sensitivity is discussed in previous chapter in detail. Steps required to reduce the noise were carried out very carefully and it is fairly evident in Figure 3.2 that highest noise is 1.3mV (1.3mV/4.02V which is 0.032%!) with average noise of 0.6mV. These results are very encouraging as to achieve such a low noise using amplitude demodulation is quite challenging.

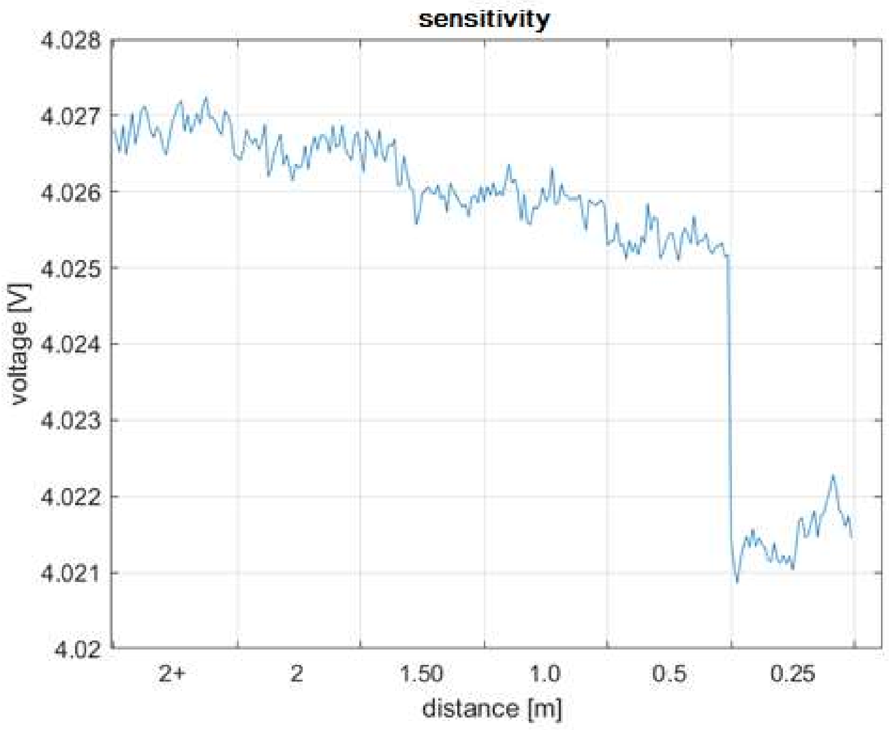

Figure 3.2: Sensitivity test with person standing at different distances (noise level 0.032%)

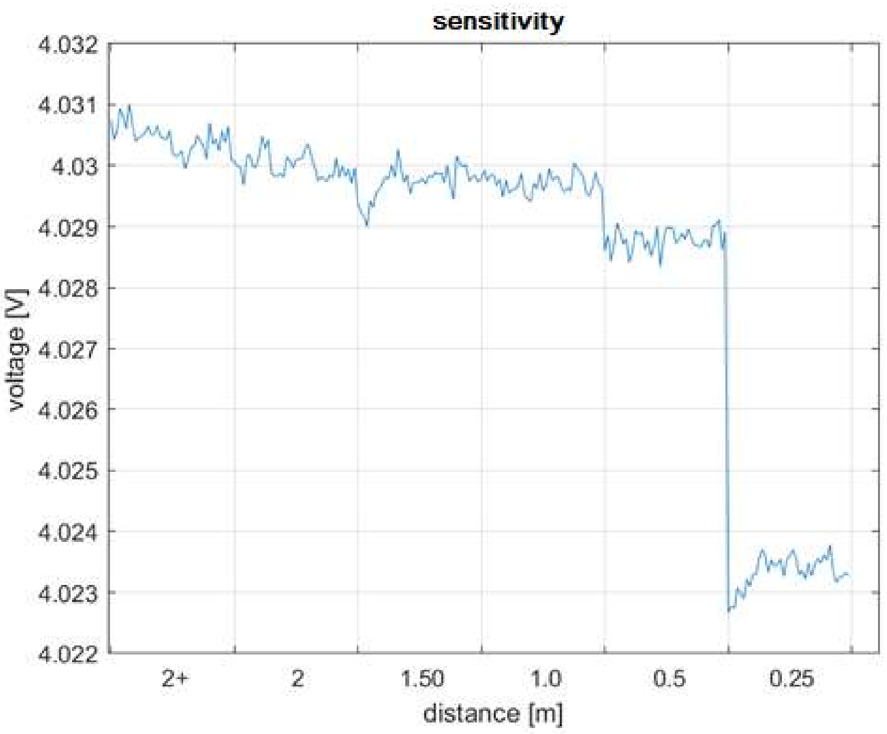

Figure 3.3: Sensitivity test-2 with person standing at different distances (noise level 0.031%)

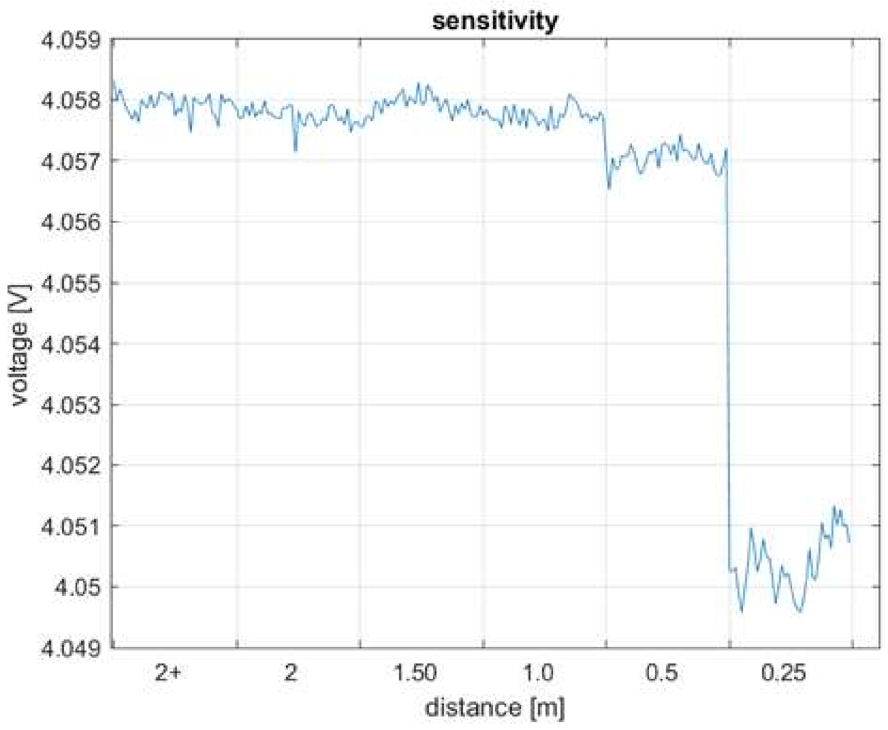

Figure 3.4: Sensitivity test-3 with person standing at different distances (noise level 0.032%)

Figure 3.3 and Figure 3.4 show other sensitivity test on another day whose average noise is even lower than of Figure 3.2. Looking at these results noise is still visible for which I can say there is electrical wiring in floor, ceiling and walls which are also a source of electromagnetic noise. Other than that the person is breathing and moving slightly which is also source of fluctuations. Difference between lowest value and highest value is nearly 8mV. Moreover, if we consider mean value of voltage during one specific distance we can say that sensor can easily detect a person upto 1.5m while beyond this it cannot be considered to be accurate.

Stability

Figure 3.5 shows a stability check over five hours. For testing and verification purposes the sensor plate was replaced with a capacitor of 18pF to check and measure effects caused by plate. Very small drift is present 5mV for 5hours i.e. 1mV/hour (5mV/4.8 = 0.1%). Which can be attributed to environmental effect i.e. temperature and humidity affecting capacitance. However, this kind of environmental influence can be compensated in software by constantly reading temperature and humidity of room.

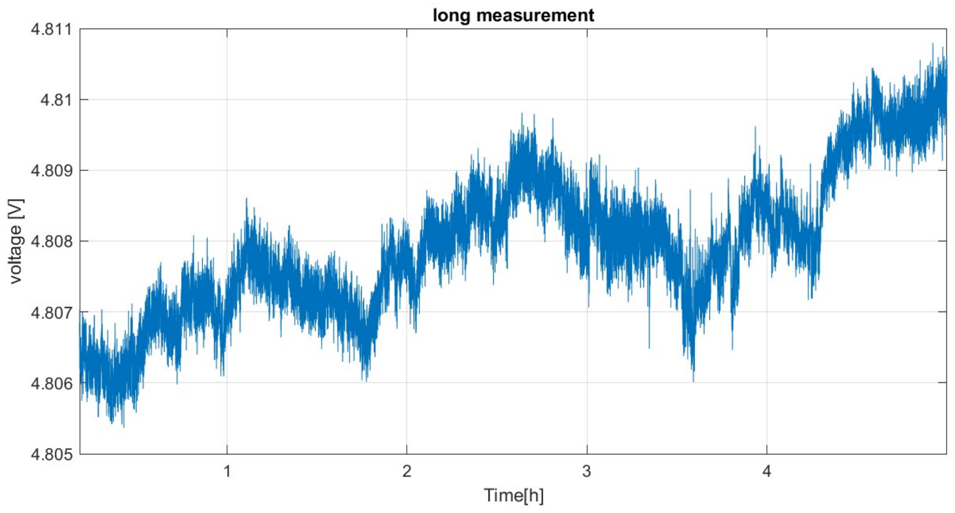

Figure 3.5: Drift measurement over five hours using lumped capacitor instead of sensor plate (5mV/4.8V=0.1%)

Figure 3.6 shows same experiment but with actual sensor plate used previously for sensitivity tests. Here it can be observed that drift is 9mV for five hours (9mV/4.884= 0.18%) slightly higher for sensor plate because of its size and open exposure to atmosphere. The negative peak nearly after one hour can be most probably due to turning off air conditioning in evening.

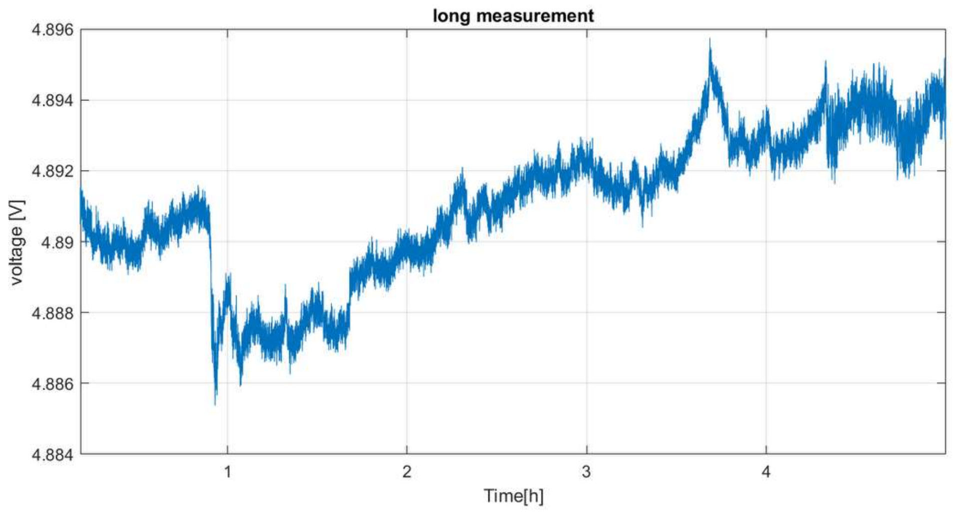

Figure 3.6: Drift measurement for five hours using sensor plate (9mV/4.884= 0.18%)

Comparison with another frontend

I compared with the experimental results of another frontend based on phase demodulation using XOR gate [9] whose sensitivity and stability are shown in Figure 3.7 and Figure 3.8 respectively.

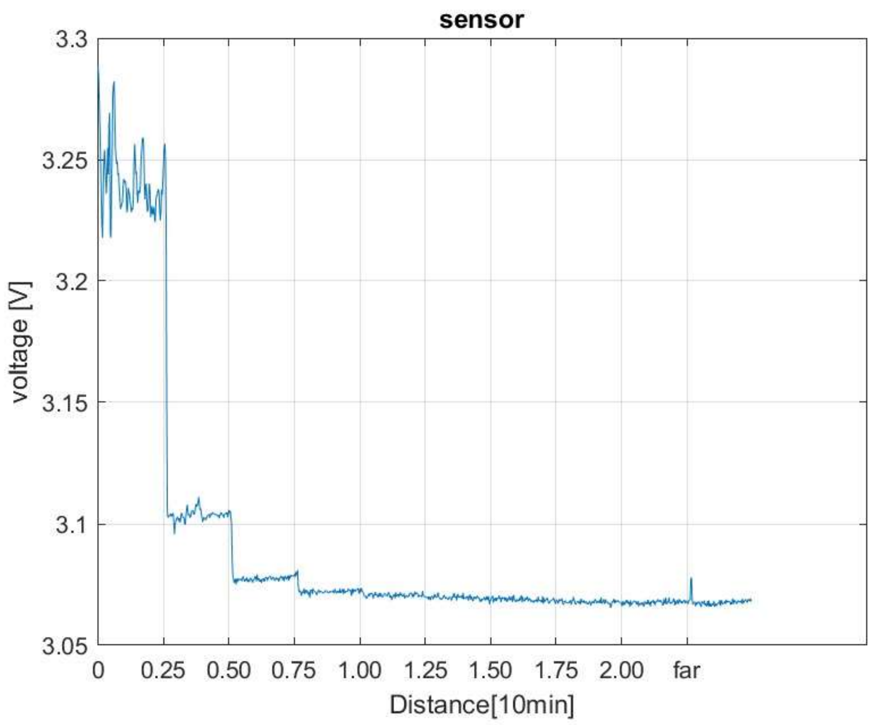

Figure 3.7: Sensitivity of phase-based frontend based on phase demodulation

Average noise level is 10mV (10mV/3.25V=0.3%) while this frontend is very sensitive and there is a big difference between highest and lowest voltage value i.e 175mV. There is clear detection upto 1.25m while response is linear afterwards.

There is a visible difference in plots of both frontends that in amplitude and phase based frontend (Figure 3.2, Figure 3.3, Figure 3.4) voltage reading decreases with distance while for XOR based frontend it decreases with increase in distance (Figure 3.7). The basic capacitive sensing method for both frontends is same. It is just a difference of voltage inversion due to use of operational amplifier in inverting configuration at last stage of signal.

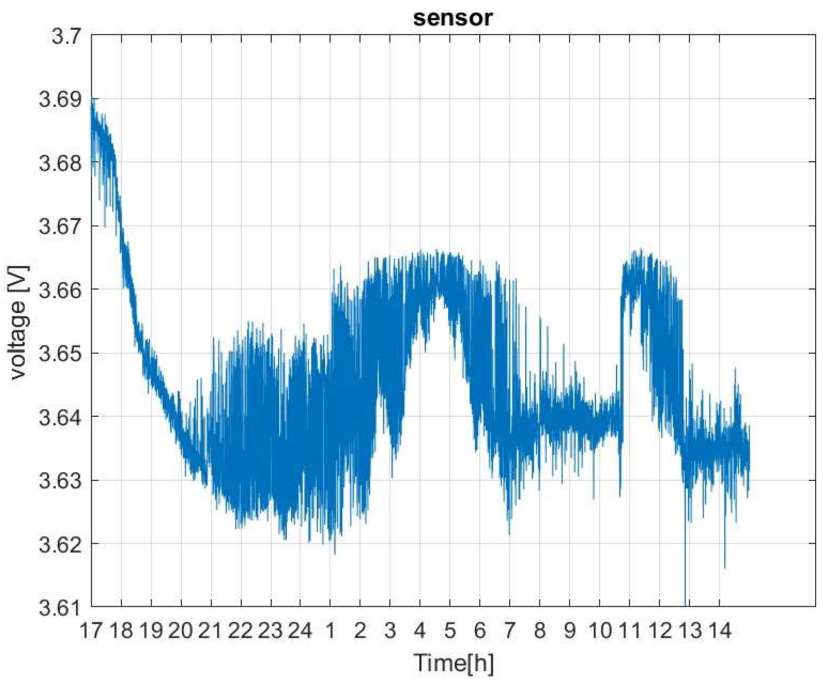

Figure 3.8: Stability of phase-based frontend based on phase demodulation

Figure 3.8 shows a stability check for drift for more than twenty hours. It is also evident that there are human movements in the room as well as other disturbances. But we extract some information for comparison. If we neglect initial voltage fall for and consider only from 21 to 14, we come to know that maximum variation in voltage is about 40mV (40mV/3.6V =1.11%).

Table 3-1 shows a brief comparison of both frontends. SNR can be estimated as maximum change in voltage from 0.25m to far distances divided by mean amplitude of noise at more than 1metre distances.

Table 3-1: Comparison of two frontends in terms of sensitivity and stability

Experiment	Attribute	Frontend based on amplitude and phase demodulation	Frontend based on Phase demodulation
Sensitivity	Peak noise	1.3mV	50mV
	Average noise %	0.032%	0.3%
	Max. Change in V	8mV	175mV

	Range of detection	~1.5m	1.25m
	SNR	16	19.4
Stability	Max. change in V	9mV	40mV
	Drift in percentage	0.18%	1.11%

Chapter 4: Conclusion & future work

Power section

As reported in chapter 2 about noise on supply lines resulted in induction of noise throughout the circuit. So, component selection for power regulators should be done with extreme care. One must consider $V_{pk\text{-}pk}$ ripple noise both in off-load and on-load configuration. Regulators requiring large number of external components must be avoided both for simplicity and cost reduction. Regulators with multiple outputs should be selected to reduce PCB complexity.

As in this circuit four different supplies were required to work completely. This should be tried to reduce. Proposed scheme for power section was to have boost regulator that should increase voltage to a higher level. Then to use LDOs regulators to have a clean supply. All component selection must be done for those packages that are easy to solder and de-solder in lab.

Band pass filters

Second order Sallen-key band pass filters were used which were well tuned to carrier frequency but as discussed in chapter 2 they were causing drift over long period of time. If there is large drift, detection and measurement are not correct. It is suggested to use simple cascaded RC filters in future designs to cope with this problem.

Operational amplifiers

The problem with operational amplifier (OP184) is high current consumption. This operational amplifier consumes current of 1.5mA per amplifier so, they are not suitable for low power application. As this is a battery powered device current consumption should be as low as possible.

PCB designing

As I mentioned earlier that this technique of amplitude demodulation is highly sensitive to any interference whether internal or external to the circuit. So, PCB designing is very critical stage. There are different techniques to reduce interference and crosstalk which needs to be embedded into this circuit. Only surface mount components should be used. As the circuit worked on a known frequency of 10kHz, impedance control and matching are crucial. This sinusoid of 10kHz is divided into two branches, so both branches must be symmetrical and balanced to obtain better results.

Components external to PCB

Although performance of the circuitry really matters but influence of external components cannot be neglected. Power banks have some switching noise on their output. Designer should keep this in mind (as said in power section). Only shielded cables should be used for power supply and especially cable connecting sensor plate and circuit. The sensor device should be properly shielded so that there is no external intrusion of noise.

List of References

[1] Javed Iqbal, Mihai Teodor Lazarescu, Osama Bin Tariq and Luciano Lavagno, “Long range, high sensitivity, low noise capacitive sensor for tagless indoor human localization” IEEE ACCESS.

[2] M. Youssef, M. Mah, and A. Agrawala, “Challenges: device-free passive localization for wireless environments", in Proceedings of the 13th annual ACM international conference on Mobile computing and networking, pp. 222-229, ACM, 2007.

[3] T. Kivimaki, T. Vuorela, P. Peltola, and J. Vanhala, A review on device-free passive indoor positioning methods", International Journal of Smart Home, vol. 8, no. 1, pp. 71-94, 2014.

[4] T. Grosse-Puppendahl, C. Holz, G. Cohn, R. Wimmer, O. Bechtold, S. Hodges, M. S. Reynolds, and J. R. Smith, “Finding common ground: A survey of capacitive sensing in human-computer interaction", in Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, (New York, NY, USA), pp. 3293-3315, ACM, 2017.

[5] J. Smith, T. White, C. Dodge, J. Paradiso, N. Gershenfeld, and D. Allport, “Electric field sensing for graphical interfaces", IEEE Computer Graphics and Applications, vol. 18, pp. 54-60, May 1998.

[6] Roberto Gambotto; supervisors Luciano Lavagno, Mihai Teodor Lazarescu, “Analog frontends for long range capacitive sensing for IoT indoor human localization”, 2017.

[7] Aurora Bellinaso; supervisors Luciano Lavagno, Mihai Teodor Lazarescu, “Machine learning algorithms for human localization using long distance capacitive and infrared sensors”, 2017.

[8] Pooya Poolad; supervisors Luciano Lavagno, Mihai Teodor Lazarescu, “Optimization of capacitive sensor front-end for indoor human localization and identication”, 2017.

[9] Javed Iqbal; MIHAI TEODOR Lazarescu; Arslan Arif,; Luciano Lavagno, High sensitivity, low noise front-end for long range capacitive sensors for tagless indoor human localization”, IEEE ; 2017

Appendix A: C Code

```c
/*
* GccApplication4.c
*
* Created: 09/24/2019 15:02:26 PM
* Author : s251819
*/

// this code sets up counter1 A output at 25% and B output at 75%
// using ICR1 as top (16bit), Fast PWM.
#define F_CPU  16000000UL

#include <stdlib.h>
#include <util/delay.h>

#include <stdint.h>
#include <avr/interrupt.h>
#include <avr/io.h>
#include <avr/pgmspace.h>
#include "USART.h"
#include "XBEE.h"
#include "XBEE_reception.h"

// Constants




// Serial port setup
#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL))) - 1)

// Server protocol
#define SERVER_REQUEST_OFFSET 8
#define SERVER_REQUEST_START_MEASUREMENT 49
#define SERVER_ADDRESS 0xABCD

//Local variables
//
struct payload_s {
uint32_t total_clocks_during_measurement;
uint32_t total_input_periods;
} payload;

uint8_t *TX_payload = &payload;
uint32_t total_clocks_during_measurement = 0;
uint32_t total_input_periods = 0;
//float input_signal_frequency = 0.0;



```

```

const int sinewave_length=20;
volatile uint8_t end_isr ;
volatile uint16_t sample;
volatile uint16_t sample;
char from_slave;

const uint8_t sinewave_data[] PROGMEM = {
20,
25,
29,
32,
34,
35,
34,
32,
29,
25,
20,
15,
11,
8,
6,
5,
6,
8,
11,
15,
  }; //10khz with sinewave_length=20, amp (5-35)
  //const uint8_t sinewave_data[] PROGMEM =
{20,26,32,36,39,40,39,36,32,26,20,14, 8, 4, 1, 0, 1, 4, 8, 14}; //10khz with
sinewave_length=20, max amplitude (0-40)
//const uint8_t sinewave_data[] PROGMEM =
{20,23,26,29,32,34,36,38,39,40,40,40,39,38,36,34,32,29,26,23,20,17,14,11,  8,
6, 4, 2, 1, 0, 0, 0, 1, 2, 4, 6, 8,11,14,17}; //5khz with sinewave_length=40
uint32_t total;
  //uint8_t LowBit;
  //uint8_t HighBit;
  //const uint8_t sinewave_data[] PROGMEM = {20, 26, 32, 36, 39, 40, 39,
36, 32, 26, 20, 14, 8, 4, 1, 0, 1, 4, 8, 14};


  ISR(TIMER0_COMPA_vect) {
  if (sample >= sinewave_length) {
  sample = -1;
  }
  else {
  OCR1A = pgm_read_byte(&sinewave_data[sample]);
  }
  ++sample;
  //end_isr=1;
  }




  void setPrescaler(uint8_t prescaler)
  {
  uint8_t mask = 0XF8;
```

```
 ADCSRA &=mask;
 ADCSRA|=prescaler;

 }
 void adc_init()
 {
 setPrescaler(4);
 ADMUX |= (1<<REFS0);
 ADMUX &= ~(1<<REFS1);
 ADCSRA |= (1<<ADEN);      //Power up the ADC
 }


 void adc_start()
 {
 ADCSRA |= (1<<ADSC);      // star convertion
 }


int main(void)
 {
 DDRB |= (1 << DDB1)|(1 << DDB2);
 // PB1 and PB2 is now an output

 ICR1 = 40;
 // set TOP to 16bit

 OCR1A = pgm_read_byte(&sinewave_data[0]);;
 // set PWM for 25% duty cycle @ 16bit 134
 TCCR1A |= (1 << COM1A1)|(1 << COM1B1);
 // set none-inverting mode

 TCCR1A |= (1 << WGM11);
 TCCR1B |= (1 << WGM12)|(1 << WGM13);
 // set Fast PWM mode using ICR1 as TOP

 TCCR1B |= (1 << CS10);
 // START the timer with no prescaler

        //setting TIMER0 for updating the values for PWM period

 // Set up Timer 0 to send a sample every interrupt.
 cli(); // disable interrupts
 // Set CTC mode (Section 15.9.2 Clear Timer on Compare Match)
 // WGM = 0b0100, TOP = OCR1A, Update OCR1A Immediate (Table 15-4)
 // Have to set OCR1A *after*, otherwise it gets reset to 0!
 TCCR0A = 0;
 TCCR0B = 0;

 TCCR0B |= (1 << WGM02);
 TCCR0A |= ((1 << WGM01) | (1 << WGM00));

 // No prescaler, CS = 0b001 (Table 15-5)
 TCCR0B = (TCCR0B & ~(_BV(CS02) | _BV(CS01))) | _BV(CS00);
 // Set the compare register (OCR1A).
// OCR1A is a 16-bit register, so we have to do this with
// interrupts disabled to be safe.
OCR0A = 10;//F_CPU / SAMPLE_RATE;    // 16e6 / 8000 = 2000
// Enable interrupt when TCNT1 == OCR1A (p.136)
TIMSK0 |= _BV(OCIE0A);

sample = 0;
sei(); // enable interrupts
adc_init();
XbeeUSART_init();
adc_start(); //start the first conversion and ignore it
while(ADCSRA & (1<<ADSC));
while (1)
{
//_delay_ms(10);
//sum all measurement
total = 0;
for(int i=0;i<4096;i++)
{

adc_start();
while(ADCSRA & (1<<ADSC));
total+=ADC;
}

payload.total_clocks_during_measurement = total;
//total_clocks_during_measurement;
payload.total_input_periods = 1234;//total_input_periods; THIS IS
DUMMY
XBee_TX_Request(SERVER_ADDRESS, TX_payload, sizeof(struct payload_s));

// we have a working Fast PWM
}
}
```

Appendix B: MATLAB codes

Fast check

```
    %%%%% External functions used in this script %%%%%
    % send_API_2.m --> sends request to gather frequency readings
    % rec_API_2.m --> receives freuency readings from cap-sensor nodes
    % send_US_req.m --> sends request for Ultrasound sensor's readings
 clear all;
 close all;
  clc;
%fclose(instrfind);

  minutes_of_run = 0.5;        %each sub-run
  hours = 1;                    %total run

  serialPort = 'COM4';    s = serial(serialPort);
  set(s,'BaudRate',9600);

  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  data_received = zeros(200000,1);

  fopen(s);

  start_time = clock;    flushinput(s);    flushoutput(s);

    data_index = 1 ; %array index to store US/cap sensor values
    i = 1;
  pause(1);    end_at = 60*minutes_of_run;    t0 = clock;
  j=0;



  while etime(clock, t0) < end_at

    if (s.BytesAvailable > 16)


    [address, total_clocks_during_measurement, total_input_periods] = 
rec_API_2b(s);

    if (address=="AAAC")
    if (total_clocks_during_measurement > 0)

    data_received(i)
             =(double(total_clocks_during_measurement)/64.0)*(5.0/65536.0);
                  %disp(address);
    disp(data_received(i));
    end
    i=i+1;

    end
    end
    end
    data=nonzeros(data_received);
    max_f=max(data);
    min_f=min(data);
diff=max_f-min_f


  fclose(instrfind);

```

Sensitivity

```
  %%%%% External functions used in this script %%%%%
  % send_API_2.m --> sends request to gather frequency readings
  % rec_API_2.m --> receives freuency readings from cap-sensor nodes
  % send_US_req.m --> sends request for Ultrasound sensor's readings
clear all;
close all;
clc;

%fclose(instrfind);
minutes_of_run = 60;        %each sub-run
hours = 1;                  %total run

serialPort = 'COM4';
s = serial(serialPort);
set(s,'BaudRate',9600);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

data_received = zeros(200000,1);

fclose(instrfind);
pause(8);
fopen(s);

start_time = clock;
flushinput(s);
flushoutput(s);

  data_index = 1 ; %array index to store US/cap sensor values
  i = 1;
  %pause(10);
beep;
end_at = 10;
t0 = clock;
  j=0;
  k=0;
  while j < 6

while k < 40


  if (s.BytesAvailable > 16)


  [address, total_clocks_during_measurement, total_input_periods] = rec_API_2b(s);
  if (address=="AAAC")
  if (total_clocks_during_measurement > 0)

  data_received(i) =(double(total_clocks_during_measurement)/64.0)*(5.0/65536.0);

  disp(data_received(i));

  i=i+1;
  k=k+1;

  end
  end
  end
  end
```

```

disp('MOVE');
beep;
fclose(instrfind);
pause(10);
fopen(s);
k=0;

j=j+1;

end

data=nonzeros(data_received);

max_f=max(data);    min_f=min(data);    diff=max_f-min_f

figure(1);
plot(data);
  %ylim([2.6 2.7])
  ylabel('voltage [V]');
  title('sensitivity');
grid on;
xlabel('distance [m]');

  %xticks(1:(length(data)/6):length(data)+1);
  %xticklabels({'0.25','0.50','1.0','1.5','2.0','2+'});
xticks(1:(length(data)/6):length(data)+1);
xticklabels({'2+','2','1.50','1.0','0.5','0.25'});

  %xticks(1:(length(data)/7):length(data)+1);
  %xticklabels({'2','2','1.5','1','0.5','1','1.5','2'});
  saveas(gcf,sprintf('sensitivity_%s.png', datestr(now,'mm-dd HH-MM')));
  save(sprintf('sensitivity_%s.mat', datestr(now,'mm-dd HH-MM')),'data');

fclose(instrfind);

```

Long measurement

```
 %%%%% External functions used in this script %%%%%
 % send_API_2.m --> sends request to gather frequency readings
 % rec_API_2.m --> receives freuency readings from cap-sensor nodes
 % send_US_req.m --> sends request for Ultrasound sensor's readings
 clear all;
 close all;
 clc;
 %fclose(instrfind);

minutes_of_run = 300;      %%%%%%%%%%%%%%%%%%each sub-run
hours = 1;                 %total run

 serialPort = 'COM4';
 s = serial(serialPort);
 set(s,'BaudRate',9600);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


data_received = zeros(200000,1);


fopen(s);


 start_time = clock;
 flushinput(s);
 flushoutput(s);

 data_index = 1 ; %array index to store US/cap sensor values
 i = 1;
 pause(1);
 end_at = 60*minutes_of_run;
 t0 = clock;
 j=0;




while etime(clock, t0) < end_at


      if (s.BytesAvailable > 16)




 [address,  total_clocks_during_measurement,  total_input_periods]  =
rec_API_2b(s);


               if (total_clocks_during_measurement > 0)
```

```
                 data_received(i)
            =(double(total_clocks_during_measurement)/64.0)*(5.0/65536.0);

  disp(data_received(i));

                 i=i+1;

                 end
       end
  end

data=nonzeros(data_received);

  max_f=max(data);
  min_f=min(data);
  diff=max_f-min_f

  figure(1);
  plot(data);
  %xlim([1100 45000]);
  %ylim([2.6 2.7])
  ylabel('voltage [V]');
  title('long measurement');
  grid on;
  xticks(1:(length(data)/5):length(data)+1);
  xlabel('Time[h]');
  xticklabels({'0','1','2','3','4','5'});%,'6','7','8','9','10'
  saveas(gcf,sprintf('long_meas_%s.png', datestr(now,'mm-dd HH-MM')));
  save(sprintf('long_meas_%s.mat', datestr(now,'mm-dd HH-MM')),'data');

fclose(instrfind);
```