POLITECNICO DI TORINO

Tagless long/distance capacitive sensors for indoor human localizations

by

kuku Tena Nigatu

A thesis submitted in partial fulfillment for the
degree of Master of Science

in the
Department of Electronics and Telecommunications Engineering

March 2018

Declaration of Authorship

I, Kuku Tena Nigatu, declare that this thesis titled, 'Tagless long/distance capacitive sensors for indoor human localization ' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Jesus Christ is the same yesterday and today and forever."

Hebrews 13:8

Abstract

Indoor Human localization:- it's main idea is to identify the location of a person inside a house as the person operates his/her daily routines. In smart homes and assisted living environments (like health centers) knowing the location of a person is important for different services that require locational information like heating, turning on lights, opening doors or behavioral recognition and notification of abnormalities.

There are different technologies that are used for indoor human localization. Applications based on pressure, infrared RFID, bluetooth and ultrasonic sensors can be found each with its own advantage and limitations in-terms of precision and infrastructure cost. Capacitive sensing is especially interesting in the area of indoor human localization due to it's many complementary characteristics to other same kind application sensing systems.

Capacitive sensing system work in different modes. Our interest is the loading operating mode, where only one electrode is required. The capacitance coupling between the human-body and electrode keeps varying as the distance between them is varied. The output capacitance interfaced with data acquisition and localization algorithm is used to infer/estimate the position of the person.

This thesis tries to optimize an ongoing project:- tag-less human indoor localization mainly by working on two topics:

The first is to characterize commercial ultrasonic based indoor navigation systems. To see if they are dependable to use them as a reference positions (real time) data input for training and checking the localization algorithm. In previous experiments synthetic data were used as a sensor readings to infer the location of the person under tracking.

The second aim of this project is optimizing frequency measurement as a proxy of capacitance measurement for ATmega328p micro-controller: by reducing the time taking to measure frequency (which in turn increase speed and reduce power consumption) and by increasing the resolution so that small (fractional) changes of frequency is taken in to account. Because at long distance the change in capacitance is very small which leads to a small changes of frequency even when the person makes a large movement.

In characterizing the ultrasonic based indoor navigation systems, the analysis of the experiments/tests shows that the accuracy was in the expected range.

However some spikes were observed. This could be because the experiment is conducted in a lab environment where there are many instruments that could interfere with the ultrasonic sensors. The way the sensors are installed is also an important factor in getting better accuracy (a line of sight and a specific amount of distance between adjacent beacons is required). Moreover the accuracy of the sensor is affected by temperature and humidity variations. If the sensor is tested in controlled enviroment by avoiding the listed limitation, better accuracy can be obtained

In designing the frequency meter, an algorithm is developed for measuring the frequency from capacitive sensor nodes by counting the exact full periods of the signal being measured (which is correlated with the capacitance between human body and the sensor, converted using a relaxation oscillator) for 40ms.

The output frequency from the relaxation oscillator is measured by using ATmega328p micro-controller. The accuracy of the frequency measuring firmware around 100kHz is 0.305Hz. Compared to the previous measuring method which took one second, in this system higher speed and accuracy is achieved.

Acknowledgements

First and foremost I want to thank God for his guidance and provision throughout my life. I am a living witness of his mercy,love and kindness.

I would like to express my deep appreciation and gratitude to my supervisor: Prof. Mihai Teodor Lazarescu and co-supervisor Prof. Luciano Lavagno for their support, guidance and useful comments through out the learning process of this thesis. I would also like to thank Mr.Osama Bin Tariq for sharing resources and his knowledge in helping me whenever i faced diffculties doing the project.

Finally, I would like to thank my loved ones: family, friends and especially my husband who has supported me during my stay in politecninco by encouraging me and looking after my precious Shalom relentlessly.

Dedicated to my family.

Contents

List of Figures

Abbreviations

WSN	**W**wireless **S**sensors **N**network
US	**U**ltra-**S**ound
NN	**N**neural **N**etworke
CPU	**C**entral **P**rocessing **U**nit
RFID	**R**adio **F**rquency **I**dentication
MCU	**M**icro-**C**ontroller **U**nit
IPS	**I**ndoor **P**ostioning **S**ystem
TDOA	**T**ime **D**ifferenc **O**f **A**rrvial

Chapter 1

Indoor Human Localization

1.1 Introduction

In smart homes and assisted living environments (like elderly people living alone in their homes) knowing the location of a person is important for different services that require locational information (to infer their typical daily behavior and notify abnormal situations which can be due to illness and may require assistance).[1]

For localizations and identification wireless sensor networks (WSN) are the most common adopted technologies. Due advancements in the areas of Information and technology (ICT) and Internet of things (IOT), it is now possible to have a low cost and large scale networked sensors that enable us measure different properties (like sound, RF signals, heat emissions), change this properties to positioning information and use cheap high speed Internet communications to broadcasts the informations over the networked system.[2]

Wireless sensor networks use many nodes installed with a reasonable distance range and measure phenomenas like temperature, humidity, pressure or other parameters according to the need for the application. Wireless network sensors can be implemented in schools, military, disaster management, animal tracking, weather monitoring and etc. [2]

Furthermore wireless sensor networks are applied to track/position human body inside a house for the purpose of making living safer and easier.[2]

Indoor Human localization:-it's main idea is to identify the location of a person inside a house as the person operates his/her daily routines. Which is important to assist elderly people, initiate alarm, turn on lights and to do the likes.

Ideally, for practical purpose indoor human localization sensors are required to be passive, tag less, privacy aware with low maintenance and deployment costs.[3]

Passive sensor:- is to mean the person is not required to actively interact with devices like refrigerator or remote-control is connected with sensors in order to be tracked.[3]

Tag-less:- Most people especially the elderly prefer not to carry the sensors/have objects attached to their body. There is also high possibility of forgetting to carry the instruments.[3]

Privacy-Aware:- Human localization should respect the privacy of the person under tracking. For example most people are not willing to be tracked using camera based sensing systems.[3]

Low maintenance and deployment costs:- For wide adoption it is highly crucial that the localization system requires low operational, maintenance and deployment cost.[3]

According to [1] there are many kinds of indoor human localizations sensors that use different techniques. Pressure sensors, infrared sensors, ultrasonic sensors, and capacitive sensors are some of passive sensors that are used to localize a person and will be discussed here highlighting their advantage and limitations.

Pressure sensors:- These sensors are arranged in a group of load cells and installed under the floor surface has accuracy of 5cm. Moreover if the weight of a person is known the system is accurate to localize and identify a specific person. The limitation with pressure sensors is that their installation cost is high and requires a lot of space which may not be available.[1]

Pyroelctric infrared (PIR) sensor:- are widely used for indoor localization application. These sensors requires a clear line of sight and are affected by heats sources that arise from light bulb and stove which results in false detection. Since they are affected by multi-path interference the number of senors required to be installed increases which in turn affect the installation and maintenance cost.

Video and Thermal imaging:- These techniques are very accurate to localize human body in indoor applications. The drawback of these system is: high consumption of power, expensive and rises the question of privacy.

There are also other kinds of indoor human localization sensors that requires the person to carry them all the time such as RFID, bluetooth, smart-phones with inertial sensors, GPS and microphone based sensors. Such systems are not effective in case of a continuous tracking is needed as the person might forget to carry the instruments.

Capacitive sensing systems:- have been exhaustively studied due to their low power consumption, low maintenance and installation cost and no requirement of the person to carry the instruments (the human body is treated as a capacitor plate side) which makes capacitive sensing system highly preferable for indoor human localization. In the next section the working principle of capacitive sensors will be discussed as it meets most of the requirement of practical application of indoor human body localization.[4]

1.2 Capacitive sensing

Researchers and applications witness that capacitive sensing has become the way for interacting with devices. For instance capacitive sensors are employed in touch screens, mobile phones, touch-pads and laptops.[5]

Since capacitance exist between objects and human-body, information can be derived by measuring the level of the capacitance coupling that exist between conductive objects including people.[6]

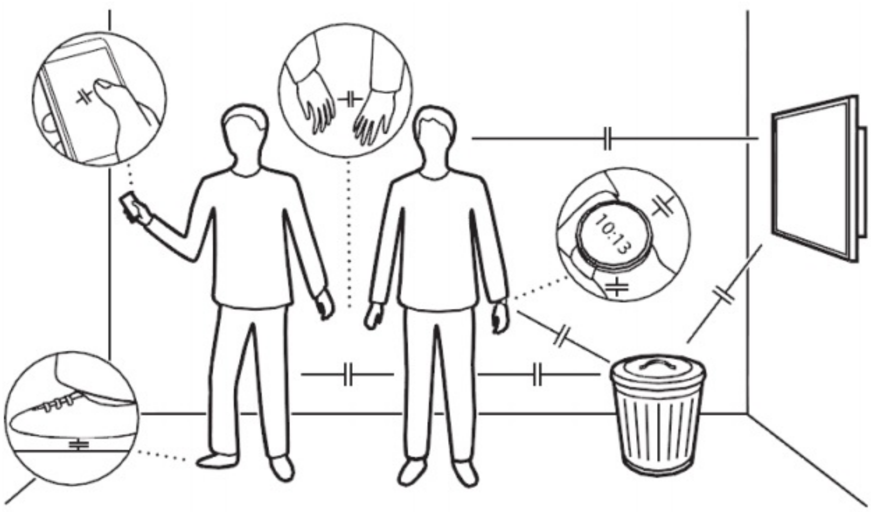


FIGURE 1.1: Capacitance and environments

From the above figure it can be easily shown that the level of capacitance coupling between objects can be measured and drive out different information like proximity, touch or deformation of objects and take different actions accordingly.[6]
The following figure models the interaction of capacitive coupling between objects (here referred as the transmit and receive electrode) and the human body. Capacitive sensing is based on measuring capacitive coupling between the electrodes

which is affected as the human body interfere with the coupling fields.[6]

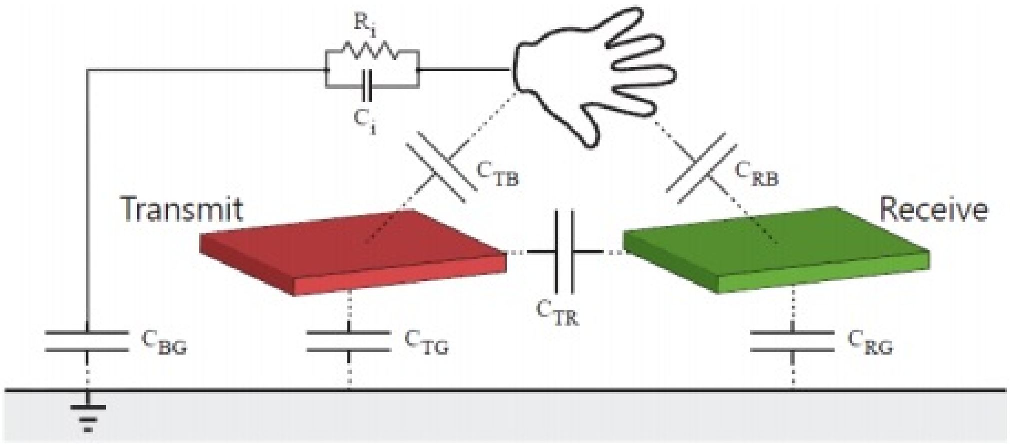


FIGURE 1.2: Modeling capacitance interaction between objects and human body

1.3 Capacitive sensing for indoor human localization

Capacitive sensing system is widely used to detect motion and location of a person in indoor application. It can be used alone and can also be combined with other sensors such as pyroelectric infrared (PIR). As described above Capacitive system has a complementary behavior and many more additional advantages as compared to other kinds of sensors in area of indoor localization.[1]

Capacitive sensors operate in four different modes. Shunt, Transmit, Receive and Load mode operations. To operate in the first three (shunt, transmit and receive) modes requires a minimum of two galvanized capacitor plates which increases the cost of deployment and resulting a complicated system.[1] Except for loading mode each mode can operate in active or passive sensing mechanism. "In active capacitive sensing, a known signal is generated on the transmit electrode, capacitively coupled onto the body part, and then coupled into the receive electrode. The presence and movement of the body part can be sensed by measuring the strength of the signal coupled onto the receive electrode. Passive sensing systems rely on existing-external or ambient-electric fields which are passively sensed".[5]

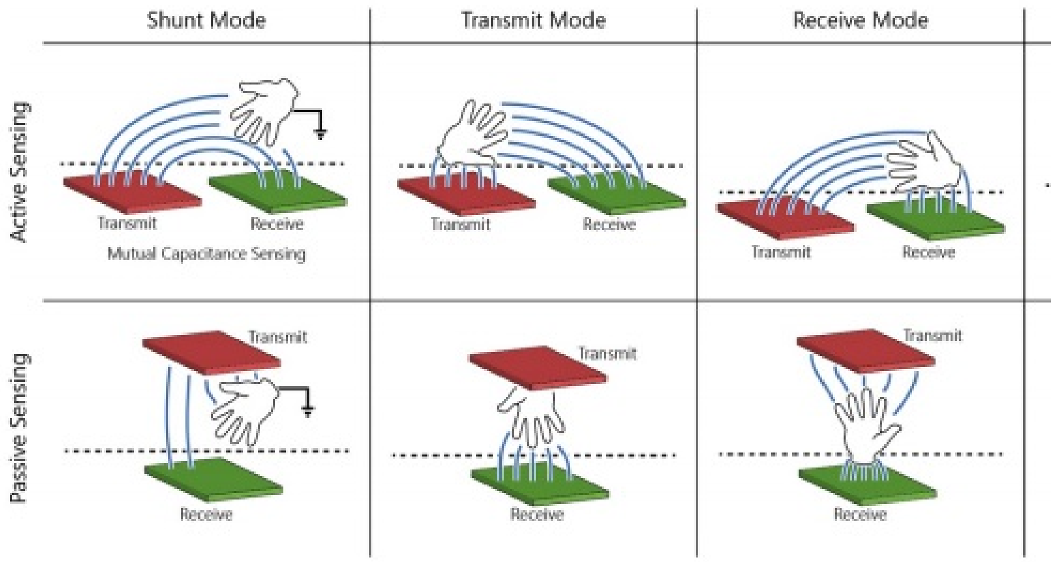
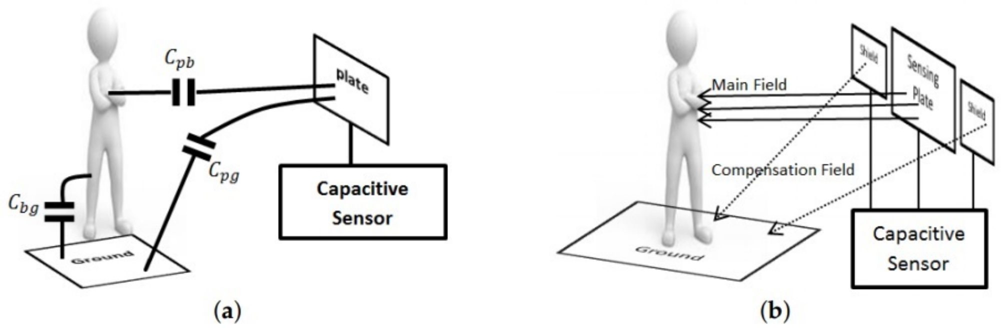


Figure 1.3: Capacitive sensing techniques can be divided into four operating modes: loading, shunt, transmit and receive. Except for loading mode, each mode may be implemented using active or passive sensing

In the fourth case (load mode operation) needs only one sensor plate while the person being tracked is taken as the other capacitor plate. Load mode configuration reduces deployment cost but it has a sensitivity proportional to the plate size used.[4]

Now the application interest is narrowed to load mode operation. The challenge will be to address or propose a solution that extends the sensitivity range and accuracy of movement detection.

The researchers at polito have designed a system that has a higher sensitivity range using a load mode operated transducer combined with data acquisition methods. The designed system meets almost all the constraints (low-power, low-cost, tag-less and privacy-aware) indoor human localization systems.[4].



(a) Main sensor capacitances in load mode: plate-body (C_{pb}), plate-ground (C_{pg}), and body-ground (C_{bg}); **(b)** Use of compensation fields for short-range load-mode capacitive sensors to reduce C_{pg}.

FIGURE 1.4: Load-mode-operation

1.3.1 Capacitor sensor

Basically, capacitance is defined as the ratio of the change in an electric charge in a system to the corresponding change in its electric potential and is given by[7]:

$$C = \frac{Q}{V}.$$

Capacitance between two parallel conductive plates separated by a distance (d) where d is much smaller than $\sqrt{A}$ is given by:[6]

$$C = \frac{\in A}{d} = \frac{\in_0 KA}{d},$$

$$\in_0 = 8.854x10^{-12} F/m.$$

$\in_0$ is permitivity of free space and K is the relative permitivity of dielectric material. K=1 for free space and is always greater than 1 for all other medias.
Charge stored between the plates is dependent on d, distance between the plates, A, area of the plates (plate size) and on $\in$, the relative permitivtiy of the dielectric

material between the plates.

For $d >> \sqrt{A}$ as in the case of Long range human localization applications sensor capacitance is proportional to $\frac{1}{d^n}$ with n typically between 2 and 3. As distance increases there is a decaying of resolution in the output capacitance.

For indoor localization, as the distance between the transducer and the human body varies the capacitance also varies. The change in capacitance is changed to frequency using a 555 astable multivibrator oscillator. Capacitance to frequency converter circuitry is shown on fig1.5.[4] The frequency is given by:

$$\text{Frequency} = \frac{1}{0.7(R_1 + 2R_2)C} \tag{2}$$

With $R_1 = 200\ \text{k}\Omega$ and $R_2 = 560\ \text{k}\Omega$, the oscillation frequency was around 70 kHz for a 4 cm × 4 cm plate.

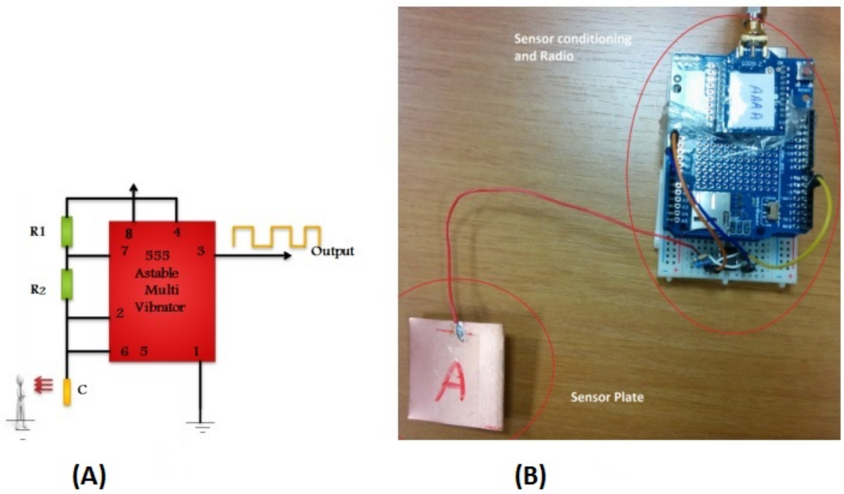


FIGURE 1.5: A) Capacitance to frequency conversion circuit, B)senor node with data acquisition module

On fig1.5 some of the measurement results of for different plate sizes is presented.

Different digital filtering techniques are used to remove the noise that affect the measured capacitance. Finally localization algorithms are applied to get/estimate the position of the person under tracking.[4]

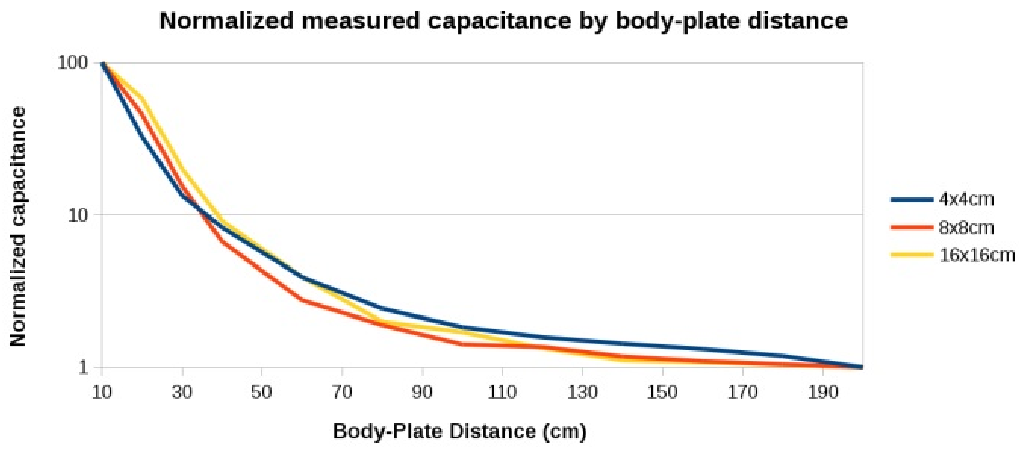


FIGURE 1.6: Normalized Output capacitance for different plate sizes and Body-plate distances

1.3.2 Advantages of using capacitor sensors

The advantages of capacitive sensing from[5] are:

No line of sight is needed:- if the distance between sensor and non conductive material is kept constant the electrodes can be installed on walls or in plastic cases. **Efficient data acquisition:** the output data can be processed using microcontroller. **Cheap Hardware:** the sensors can be prototyped easily using cheap material, it just needs conductive plates and simple circuit design. Moreover, capacitive sensors are small in size and also can be produced in large size depending on the need of the application. With efficient data acquisition the sensors could also work at high update rate.

1.3.3 Challenges of using capacitor sensors

Some limitation of capacitive sensors as mentioned on[5] are:

Affected by noises: the out put capacitance is one dimensional value and is affected by noises coming from conductive object and the environment. For example for the application of indoor localization the dielectric material is air, which is affected by variations in temperature and humidity or interference of conductive object between the electrode and human body. Another limitation of capacitive sensor is **Ambiguous output**. At short distance, changing the size of the object has a little impact on the output capacitance, which stays in the same magnitude order. Consequently it is not possible to distinguish between small objects at short distance or big objects at greater distance. There is also **Decaying of resolution** as distance increases. After few centimeters, the relationship between capacitance and distance is given by:

$$C = \frac{\in kA}{d^x}, \text{ where x is between 2 and 3.}$$

Therefore, due to ambiguous output and the effects of noises, additional information is required to interpret the output from capacitive sensors.

1.4 Localization using sensor data

Positioning a moving object analytically is possible using the methods like triangulation and trilateration when the distance measurements are obtained. But when measurements of distance are not constant (affected by noises and fluctuate a lot), it becomes beyond the scope of analytical methods to position/track a moving object in three coordinate system.[8]

As it can be observed from the above figure, Localization of an object of a noisy measurement with analytical methods could result in false/erroneous tracking (it could be any where in the dark region).

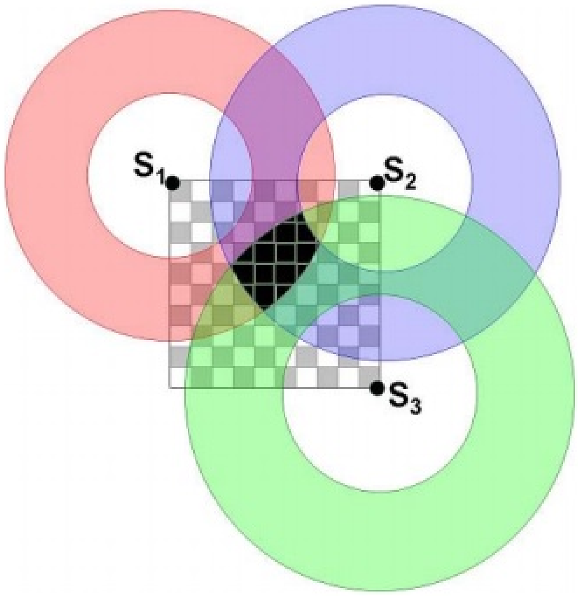


Figure 1.7: Noisy-distance-problem

So, to avoid this we need a system that is able to take the effects of noise and fluctuation into account and position the object being tracked accurately in three coordinate system. For this application neural networks are currently being widely considered showing a promising result in the area of localization/tracking a moving object.[8]

The way how the sensors operates to measure distance affect the level and source of noise. Typically stationary beacons are installed on walls (ceilings) and measure the distance from mobile nodes. some sensor uses the strength of the received signal (RSS)-RF signal based measurements and others use the acoustic wave and combination of both RF signaling and acoustic wave reflection. In both cases it is possible to calculate distance (position of mobile nodes) by sending the RF or sound wave to the nodes and measuring the time it takes for the signal to bounce back from the target.

A significant advantage of neural networks is that prior knowledge of noise distribution is not required. Neural networks compensate for noise and can be trained to associate measured positions (in our case measured frequencies) with actual locations (reference x,y,z positions).[8]

1.4.1 Neural Network (NN)-based localization

Neural Networks:- inspired by human brain (neuron) connection system, a computing system has been implemented for different application domains that learns progressively and improve over time by taking examples into consideration without task specific programming. For instance if we take image recognition, just like human brains detect distorted images without a problem neural networks could also learn to detect images in the presence of noise.[9]

To implement a neural network, it requires input layer, one or more hidden layer and output layer.

The basic element of neural network is neuron. Each neuron implements a specific function and outputs to the next layer neuron connected by using synapses that simply multiply the output from a neuron by a bias or weight.[10]

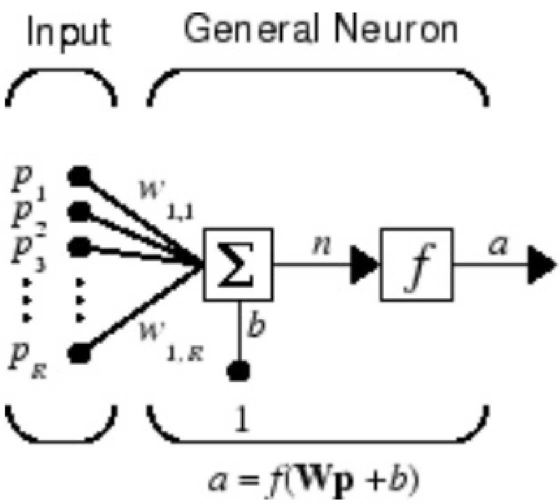


Figure 1.8: An element of neural-networks: neuron

As it can be observed from the figure each input neuron p is weighted by W, take the total sum by adding a bias and is given to a transfer function-f to provide a proper output. The transfer function could be sigmoid transfer functions: logsig and tansig or linear transfer function as purelin.

Combining many neurons and including many layers will result to a neural network. In the following pictures non linear functions are used: which helps the system to learn non linear relationships.[9]

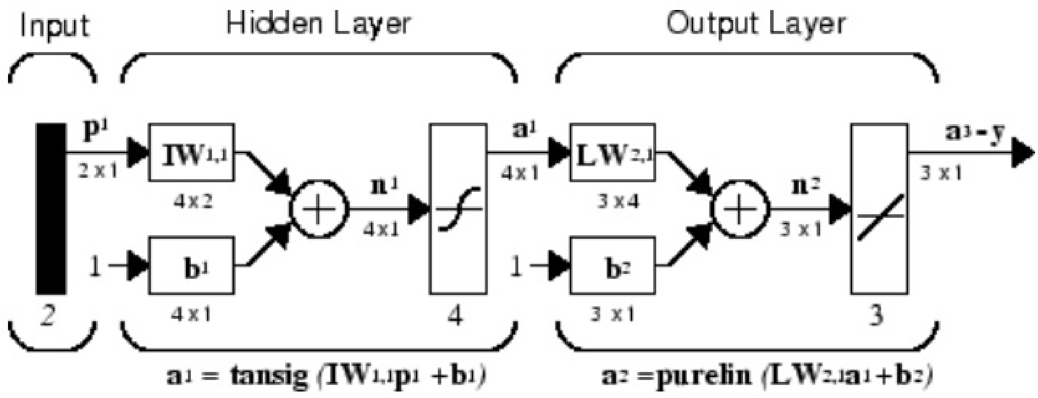


FIGURE 1.9: Neural-network Architecture: implemented example

Major steps followed to implement Neural networks are Collect data, Create the network, Configure the network, Initialize weights and biases, Train the network and finally Validate the network (post- analysis). The best output is reached when target and output are the same or as closest as possible. For this to happen different optimization techniques are used, but also the input data, that is used to train the network is very important. So the reference input data needs to be accurate or dependable so that it would let the network recognizes patterns precisely.[11]

1.4.2 Experimental data for NN training and testing

Neural network based localization algorithms, 1st they are trained by giving them reference positions that are measured by Ultrasonic sensors (using indoor navigation system from marvel mind robotics) and associate those positional values with the measured values of the frequency from the capacitive sensors. So later after training, by removing the reference positioning system the neural network will be able to tell the position of the person solely depending on the output of capacitive sensors.[12]

Previously, synthetic data were used for experimental purpose to train and test the neural networks.[12]

1.5 Previous work

The researchers here at polito have developed a long range capacitive based indoor localization system. The experiment has been conducted in 3X3m room where four sensors are installed on the wall as shown as on fig1.10:- the 3D model of the room.

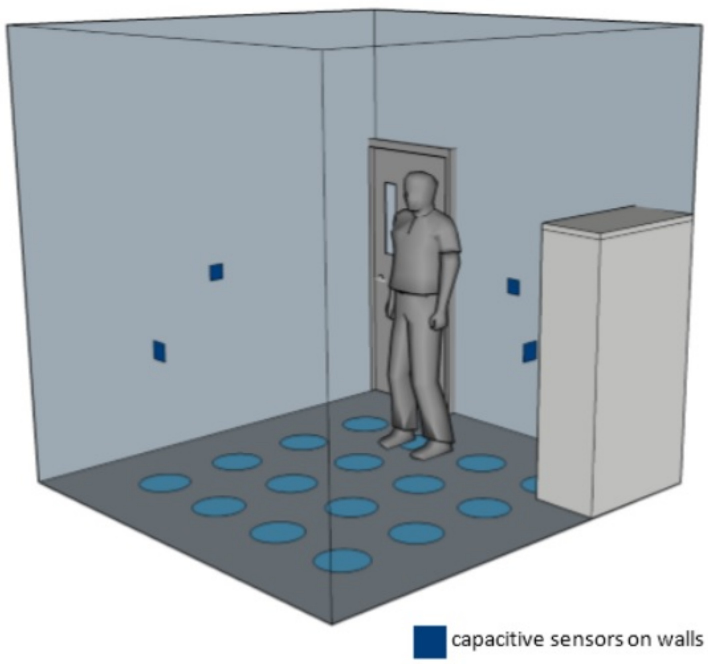


FIGURE 1.10: Capacitive sensors deployment in the 3m 3m room

Main building blocks of the system are presented on fig1.11. It has a front end that encompasses a transducer (the capacitance plate), a relaxation oscillator that changes capacitance in to frequency, an MCU (ATmega328p micro-controller) to count the frequency and send the frequency over radio (XBEE shield) module. Additionally the system has base station (Data Acquisition) which makes the back end of the system, receive the frequency from sensor nodes via a radio (XBEE shield) module, filters out noise (digital filters are implemented), use a machine learning algorithm to associates measured capacitance/frequency to reference (actual input data) positions and output the position of the data.[12][4]

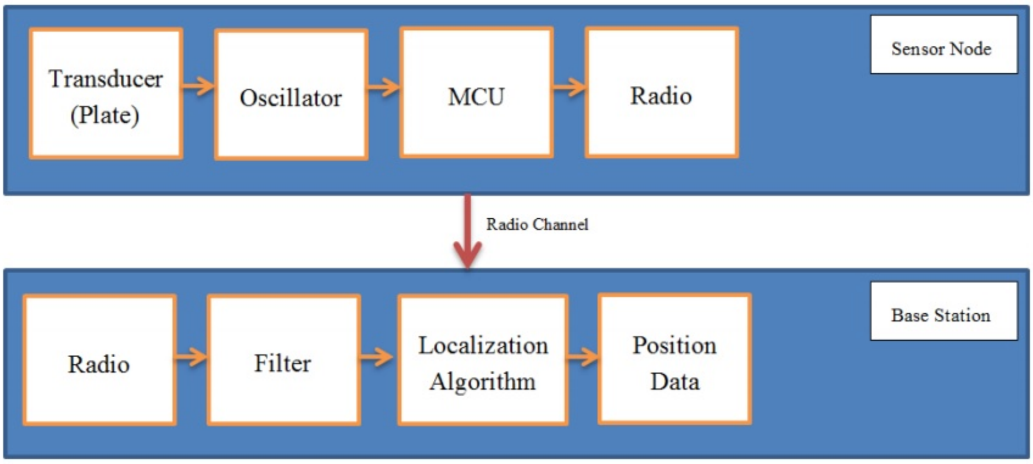


Figure 1.11: Main building blocks of localizing system with sensor node and base station

To conduct the experiment, they tried to setup an environment that simulates the actual living home scenario. As shown in the figure (fig2.12) below there are metallic objects like door, refrigerator that could interfere with the coupling field between the sensor node and human body to be detected. During conducting the experiment a person was asked to move on 16 fixed position in a specified pattern. It was needed to repeat the experiment several time to get significant analysis and also see the consistency of the system in different experiments.

The results obtained were good. Using 16x16cm capacitance plate size, distance can be inferred at 2m and the average error was 0.2m.[12]

1.6 Main contribution of this thesis

Major goals of the project is to improve the accuracy of sensor reading, increase sensitivity range (increased distance), reduce the plate size and reduce noise. To implement this it requires to optimize each module (building blocks) of the system.

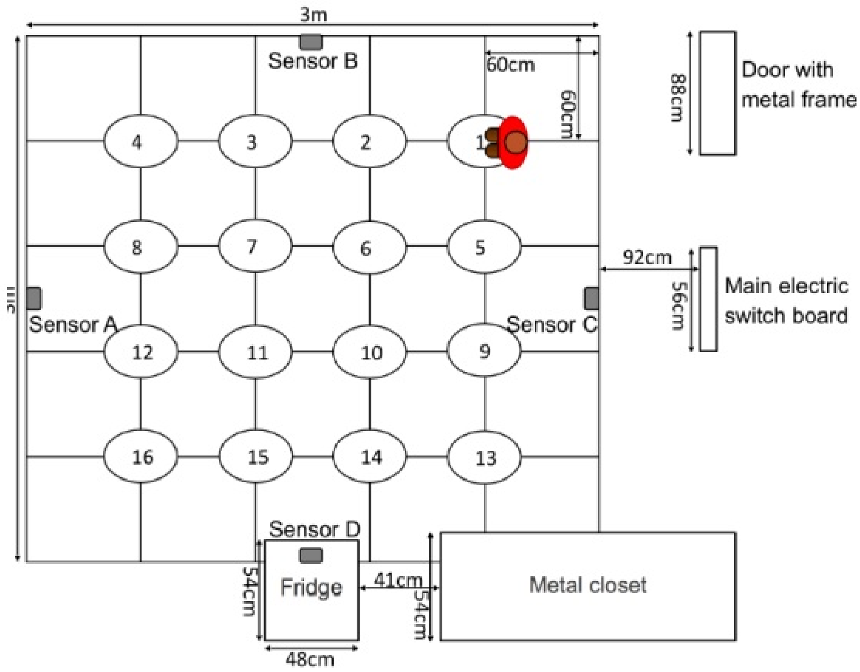


FIGURE 1.12: Room layout, sensor placement, room positions for the 16 position measurement

The focus of my thesis will be to give an input towards the optimization of the project by working on two areas:

The first is to characterize commercial ultrasonic sensors, see if they are dependable to use them as a reference for person position. To label sensor data for localizing algorithm. Previous experiments used synthetic sensor data labeled with person position to train and check the inference accuracy of the location of the person under tracking.

The second aim of this project is optimizing frequency measurement firmware for ATmega328p micro-controller, by reducing the time taking to measure frequency (which in turn increase the speed and reduce power consumption) and reducing the discretization errors so that small (fractional) changes of frequency is taken in to account because at long distance the change in capacitance is very small which leads to a small changes of frequency even when the person makes large movements

Chapter 2

Reference system based on ultrasonic sensors

2.1 Introduction

Ultrasonic sensors use sound waves to calculate the distance of a target. The sensors generate a sound wave that travel at 334m/s. The signal is received after bouncing back from the target.[13]

$$distance = \frac{speed\ of\ sound\ \times time\ taken}{2}$$

Temperature and Humidity affects the accuracy of ultrasonic sensors, errors keep increasing after certain temperature. Moreover some objects like carpet or clothes absorb the sound waves instead of reflecting them back which hinders the ultrasonic sensors from detecting them. A line of sight between the target and the sensor is required for detection. The shape object being tracked need to be considered as it could deflect the signal to another angle and not reaching back to the sensor.[13]

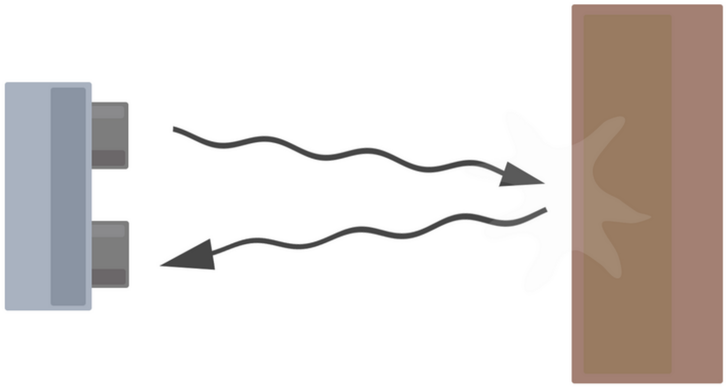


Figure 2.1: Working principle of ultrasonic based sensors

2.2 Accurate and low cost ultrasonic sensors, used for position labels

Internet of things (IOT) stimulated the development of many indoor positioning systems based on different technologies. There are Radio frequency (RF) signals based systems that require few cost of deployment with low accuracy. Whereas for UWB (ultra wide band) based on time of arrival (TOA) systems accuracy is tens of centimeters with higher infrastructure cost. Again this accuracy gets lower for Wifi based systems which is several meters and tens of meters for Zigbee and RFID (radio frequency identification) systems.[14]

Such systems cannot meet the requirements of accuracy where it is in few centimeters. On the contrary ultrasonic based applications for indoor positioning systems with traits such as low cost of infrastructure, reliability, scalability, easy penetration of wall, operating on low energy and higher accuracies makes them interesting for indoor positioning systems (IPS).[15] From the table it is easier to compare the complexity, accuracy and scope of different indoor positioning systems and see how Ultrasonic based applications standout in every aspect.[14]

Method	Accuracy	Complexity	Cost	Scope
UWB	Medium	Higher	Higher	lower
RFID	High	High	Low	High
ZigBee	Medium	Lower	Lower	Medium
Infrared	High	Low	High	Lower
Bluetooth	Low	High	Medium	High
WiFi	Lower	High	Lower	Higher
Ultrasonic	Higher	Lower	Lower	Medium

Table1: comparsion of different indoor positiong methods based on their accuracy and complexity

Ultrasonic based location systems use time of flight (TOF) measurements calculated from the velocity of sound. But the speed of sound in the air fluctuates due to the effects of humidity and temperature which will affect the measured TOF. The significant effect comes from temperature as it is depicted in the equation 2.1 given below. Where T is given in Kelvin.[16]

$$Vus = 20.05\sqrt{T}.\ \text{EQ2.1}$$

For every change of temperature in degree Celisious there is 0.18 percent change in the speed of sound leading to incorrect localization. In most of ultrasonic based localization systems temperature sensors are also implemented to take account the effect of temperature. Ultrasonic sensors use narrow or wide band each with its own advantages and drawbacks. When using narrow band signals travel longer distance with low power and the complexity of hardware is small. The limitations of narrow band is: unable to identify multiple signals. In using wide band ultrasonic (US) power and complexity increases as the signal are attenuated shortly after transmission but they give multiple access and reduce interferences from other emitters.[14]

Ultrasonic based system give us location update within some intervals which is called *frequency or update rate*. Higher update rate results in higher accuracy.[16] Among the well known ultrasonic based indoor positioning applications some of them are presented in the following sections:[16].

The cricket indoor location system: developed by MIT, composed of stationary nodes attached on ceilings described as passive-listener and active mobile node (the one's location to be estimated). The systems uses the combination of RF and ultrasonic waves to calculate the time difference arrival (TDOA) of the signal to provide position and orientation information.

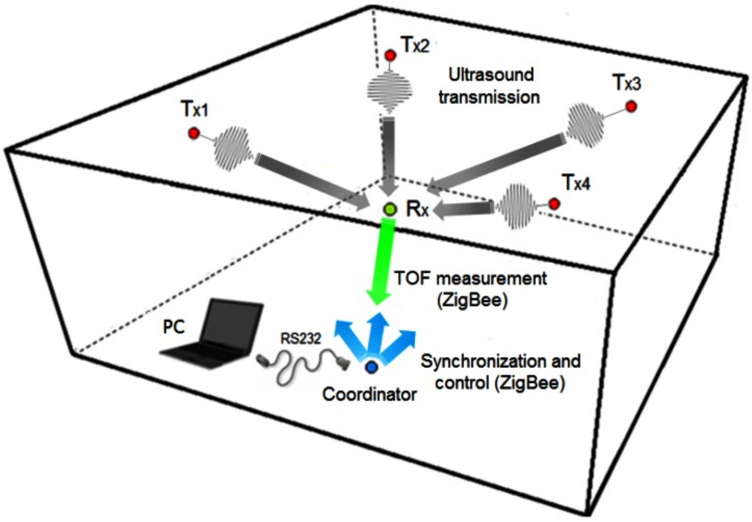


FIGURE 2.2: Configuration of the cricket positioning system. A coordinator node is connected to a PC with set of fixed nodes work as ultrasonic transmitters and The mobile node operates as ultrasonic receiver.

Each node has a temperature sensor installed with it to compensate errors caused by temperature variations. Beacons transmit their messages using RF signals and hearers (receivers) use the average temperature of transmitters and receiver for location estimation. Cricket indoor positioning system reaches an accuracy of distance measurement 4-5 cm, localization accuracy of 10-12 cm and orientation (angle) accuracy of $3^0 - 5^0$. They implement algorithms that filters out noises and incorrect readings that comes from ultrasonic noise present in the surrounding. Performance of cricket indoor positioning system decreases drastically if noise is persistent.

BUZZ indoor positioning system:- two (Synchronous and Asynchronous) BUZZ positioning systems were developed using solely narrow band communication method. Localization is attained using mobile nodes attached to the body of human by communicating timing information from main nodes to the mobile nodes.

The narrow band channel is exploited using time division multiplexing mechanism. For the sake of synchronization a wired connection of beacon to the central system is implemented which limits the practicality of the system in some situations. The system demands a minimal of four beacons to do measurements and adds a constraint on the mobile node to be placed in a specific manner at the starting of measurement.
Looking at the accuracy, BUZZ indoor positioning system gives a 4cm positional accuracy for 50 percent of the time and 10cm positional accuracy for 95 percent of the time with update frequency rate 33Hz.
Asynchronous BUZZ positioning systems targets application that doesn't require high accuracy. Signal transmission is according to internal clock of each beacon providing positional accuracy around 20-50cm. Some Advantages BUZZ positioning system is:- it has lower cost (built with lesser components) and low power consumptions.

Buzz has lower accuracy due to the neglect of temperature's and humidity's effect on the speed of sound causing higher errors in estimation of location.

Dolphin indoor positioning system:- each node has RF transmitter and receiver known as dolphin. It applies DSSS (direct sequence spread spectrum) method that enables measurement at different angles and reduces the errors caused by noises that comes from the surrounding.[17]
Compared to other US based systems, especially those with narrow band signaling where channel access is limited, dolphin has higher location update rate as it implements CDMA (code division multiple access).[17]

From the dolphin indoor positioning system architecture on fig-2.3 we can see that it consists of normal and reference nodes each with RF transmitter and receiver. The RF function as in all other cases is used for synchronization purpose. To position first a reference node send RF signal containing its location, secondly the other nodes receiving the RF signal starts counting after a while the reference node sends Ultrasonic pulse and finally the nodes receiving US pulse start computing distance and position the mobile nodes. Only few configured reference nodes are

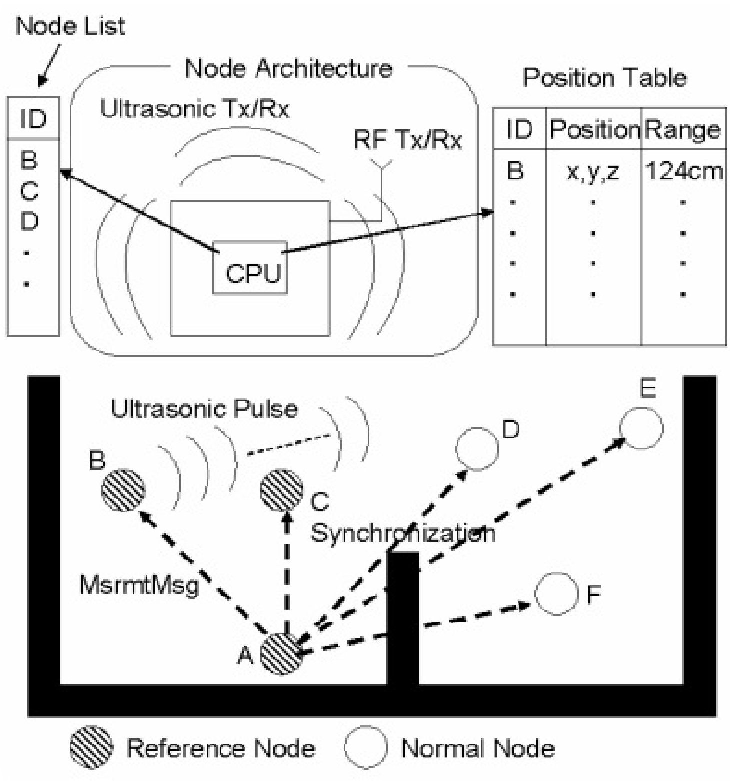


FIGURE 2.3: Dolphin positioning system

required to compute locations of other nodes.
Comparing to other similar systems the accuracy of the dolphin system even in the presence of noise is 2.3 cm which is much better.[17]

To summarize, the following comparison table of Ultrasonic based positioning system is presented. comparing to other localization systems US based localization systems provide precises positioning and their deployment cost is low.[16]

Table 2: Comparison of US systems Reviewed

System	Spreading and Channel access	update rate	Measurement method	Accuracy (cm)	Orientation (degrees)	Structure	Cost
Cricket	-	Low (1 Hz)	TDOA	10 cm	3-5	Decentralized	Low
Buzz	-	High (33 Hz)	TOA	4-10 cm	Not supported	Centralized, decentralized	Low
Dolphin	Gold Codes / CDMA	High (20 Hz)	TOA	3 cm	Not supported	Centralized	Medium
D	Kasami codes FHSS	High	TOA, AOA	1.5 cm	4.5	Centralized	High
E	Gold Codes / CDMA	High	TOA	2 cm	Not supported	Centralized	

To provide locational estimations, each US based systems needs stationary nodes that must be installed on walls and battery operated mobile nodes attached to either moving robots or human body.

Due to the noisy nature including multi-path propagation, lack of line of sight, temperature and humidity of indoor environments, there is a limitation on RF and US based IPS in estimating locations. Some systems are able to reduce the incurred limitations using different methods. Ultrasonic based systems are more affected by permanent noises that harshly degrades their performances.[16]

There are no systems that work perfectly, each system has its own limitation. To choose among systems one must first find the application domain of the sensors, and do trade off among accuracy, update rate, scalability and cost.[16][14][17]

2.3 Indoor navigation system from Marvel-mind robotics

Form the previous section it is agreeable that ultrasonic based senors are preferable for indoor positioning due to there high accuracy, reliability and low cost of infrastructure. To provide a reference real time data, that would be combined with the filtered output of the capacitive sensors and be input for the localization algorithm, a commercial sensors that operates based on ultrasonic principle interfaced with radio frequency signals in license-free band was acquired from Marvel-mind robotics.[18]

As it can be shown from the pictures, there are four stationary beacons, one mobile beacon which is called the HedgHog and a modem which will interact with mobile and the stationary beacons. The modem/router is connected with pc via USB and interact with each beacon/hedghog and display their status using a software (called the dashboard). From the base station It is possible to view/track as the a person or robot carrying the sensor moves, adjust different settings for the beacons

FIGURE 2.4: Commercial indoor navigation systems:- set of ultrasonic based beacons, modem and one mobile beacon(Hedgehog)

and view the location of each beacon. The maximum possible distance between beacons is 50m but the recommended distance is 30m apart.[18]

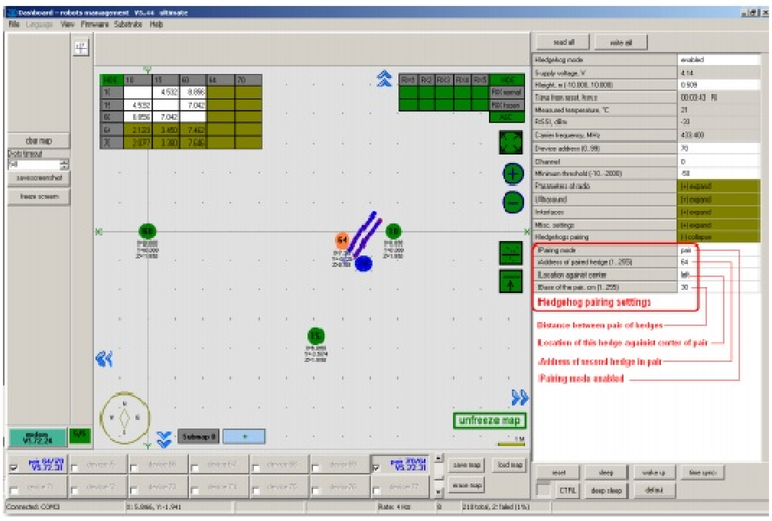


FIGURE 2.5: Dashboard: a software connected with the modem is used to adjust settings and view the locations of beacons

2.4 Characterizing the Ultrasonic sensors

Characterization is done before using an instrument in a working environment or in an experimental setup, to see if the instrument is working per the specification in the data-sheet. It is also important to understand the behavior of an instrument so that it is exploited to work in away we need it.

From the vendors and data-sheet of the ultrasonic sensors we can see that it has precision of +/-2cm (the minimum change in movement it can detect is 2cm) and sampling rate (default location update rate) is 16Hz which can be decreased to 1Hz or increased up to 45Hz. The drawback of this system is it requires direct line of sight between beacons to calculate the position of the mobile hedghog. The maximum possible distance between two beacons for accurate detection is 50m.[18]

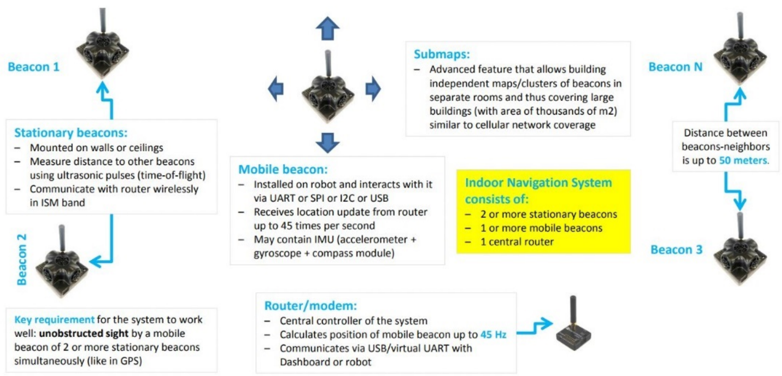


Figure 2.6: Review of indoor-navigation-system

Key capabilities:

Parameter	Technical Specifications
Distance between beacons	Reach up to 50 meters in lab conditions. Recommended distance is 30 meters (Transducer4 to Transducer4 looking straight at each other and other transducers are off)
Coverage area	- Reach up to 1000 m^2 with the Starter Set configurations - Coverage for larger territories is similar to cellular networks
Location precision	- Absolute: 1–3% of the distance to the beacons - Differential precision: ±2 cm
Location update rate	- 0.5–45Hz - Can be set manually - Depends on the distance between the mobile and stationary beacons (shorter distance—higher update rate) - Depends on the number of mobile beacons (update rate of 25Hz for 1 mobile beacon, 25Hz/2 for 2 mobile beacons, and 25Hz/3 for 3 mobile beacons) - Depends on the radio interface profile (500kbps vs. 38kbps) - Slightly depends on the number of stationary beacons—different than for mobile beacons
Power supply	Internal: LiPol battery 1000mAh - Battery lifetime depends on usage - Stationary beacon with 16Hz update rate => up to 72h (tested). - Stationary beacon with 1Hz update rate => ~72h*16 => 1 month - Mobile beacon with 8Hz update rate – 12h (tested)
	External: micro USB – recommended for permanent use
Weight	Mobile beacon from starter set: - 59 grams (including battery 1000mAh and housing and antenna 50mm) - 27 grams (bare board w/o battery)
Beacon size	Size: 55x55x33 mm (with 50mm antenna: 55x55x65mm)

2.5 Experimental setup for indoor navigation system

The main steps followed in characterizing the ultrasonic sensors:

1.Setup the equipments (installing the sensors),

2.Developing a Matlab script to analyze and do interpolation over the acquired data and

3 Do the experiment and make conclusions.

Interpolation is required because the system sends positions at unknown speed with many redundant data. By using interpolation we can assign a fixed rate for taking the samples which is also important to synchronize with the capacitive sensor nodes and for reducing the redundancy.

Setup the equipments (installing the sensors): During characterization of the indoor navigation system the four beacons were installed (put) near the walls in 6x4meter room. A person put the Hedghog on hat (for stability) and was asked first to move on 16 fixed positions (taking samples while the person is standing still in one position for about 5 second), then to walk back and forth at a constant speed on a straight line for about 1 minute and in some cases for 6 minutes.

Analyzing and interpolating over the acquired data: using Matlab scripts post analysis was done on the data from the mobile beacon which the system records in a log file (called hedghog.log and has the position of stationary and mobile beacons including time stamps).

The first analysis is done for 16 positions. After taking 20 samples at a position and calculating the standard deviation among 20 data set, the error is equal to 10-12 cm. This value is the maximum error. For most of the cases the difference between samples is near zero.

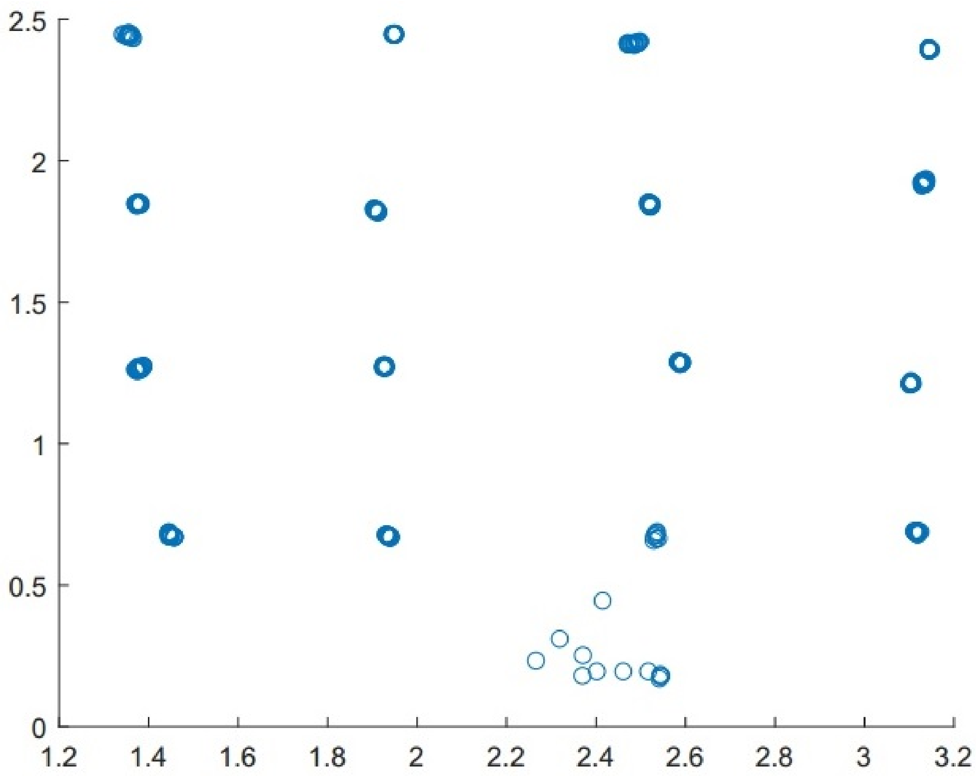


Figure 2.7: Plots: person standing on each-16 position for 5s putting on the beacon. 20 samples per position

The second analysis is done on the data taken on straight line. The data is taken by walking back and forth on a straight line of approximately 3.5m at a fairly constant speed for 5-6 minutes. For most of the trajectory path speed is constant except at turning points. Number of samples from the hedglog file=17542.

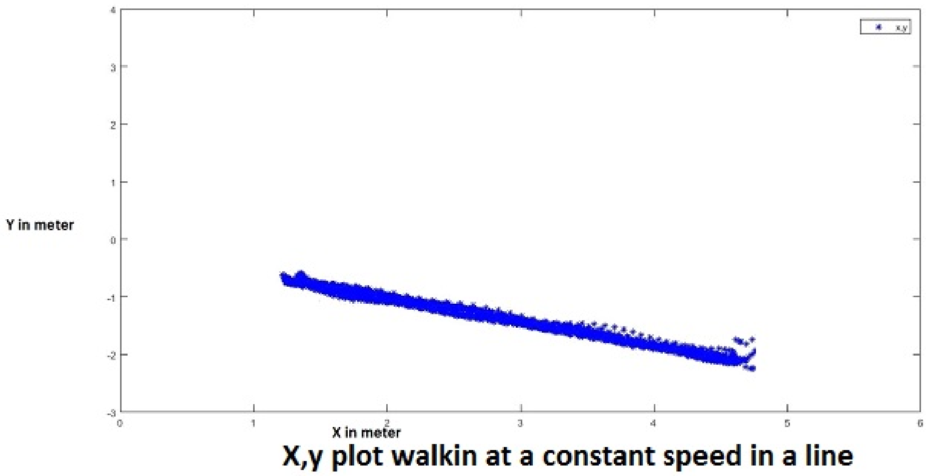


FIGURE 2.8: showing the trajectory path of the person walked by putting on the mobile beacon for about 6 minutes

From the above plot it can be observed that the samples align on st line as expected. On fig2.10, we can see x and y plotted separately with error bar (dx,dy) showing a sinusoidal wave structure since the data are taken by walking back and forth on a straight line. At round about, speed decreases and there are more samples as compared to the other parts of the line (where speed is constant).

Calculating the change in X coordinate $dx = abs(Xold - Xnew)$ and similarly the change in Y, dy, we obtain the difference between two successive samples (dx,dy). From The plots dx and dy in fig2.9 we can observe that for most of the path the difference between successive samples is nearby constant. Fig2.10 shows how the above (dx,dy plot on fig 2.9) correlates with X and Y coordinates. We can see that on the constant speed segments of the trajectory, the fluctuations in dx and dy are distributed within a few cm intervals.

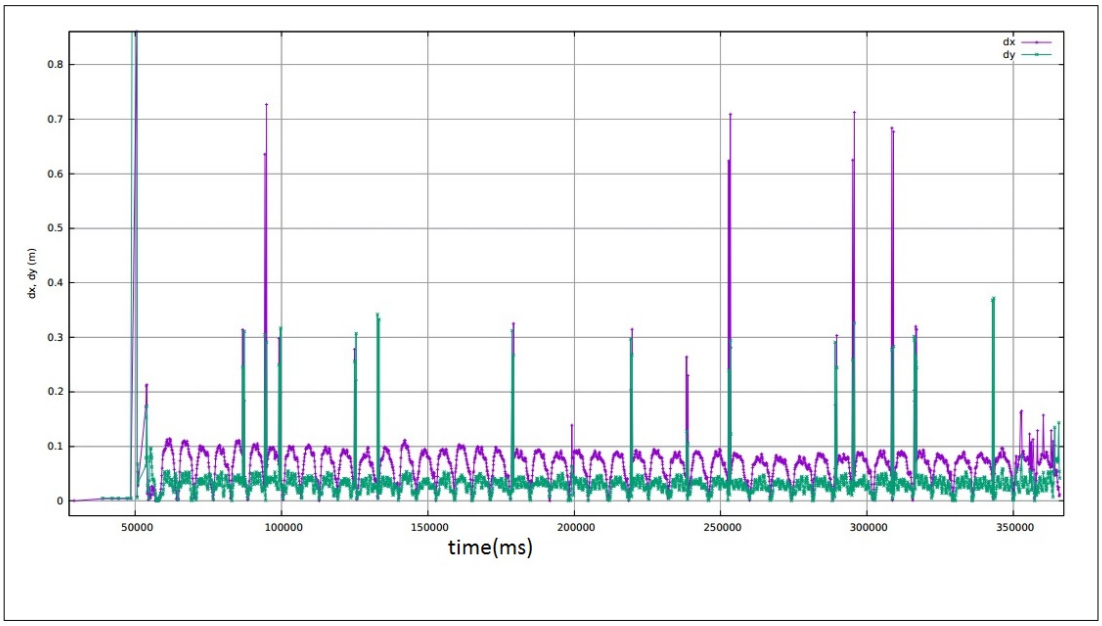


Figure 2.9: The difference b/n successive samples (dx,dy) of the person walked by putting on the mobile beacon for about 6 minutes

From the analysis of the experiment, it is proven that these ultrasonic sensors can be used as a reference person positions to label sensor data.

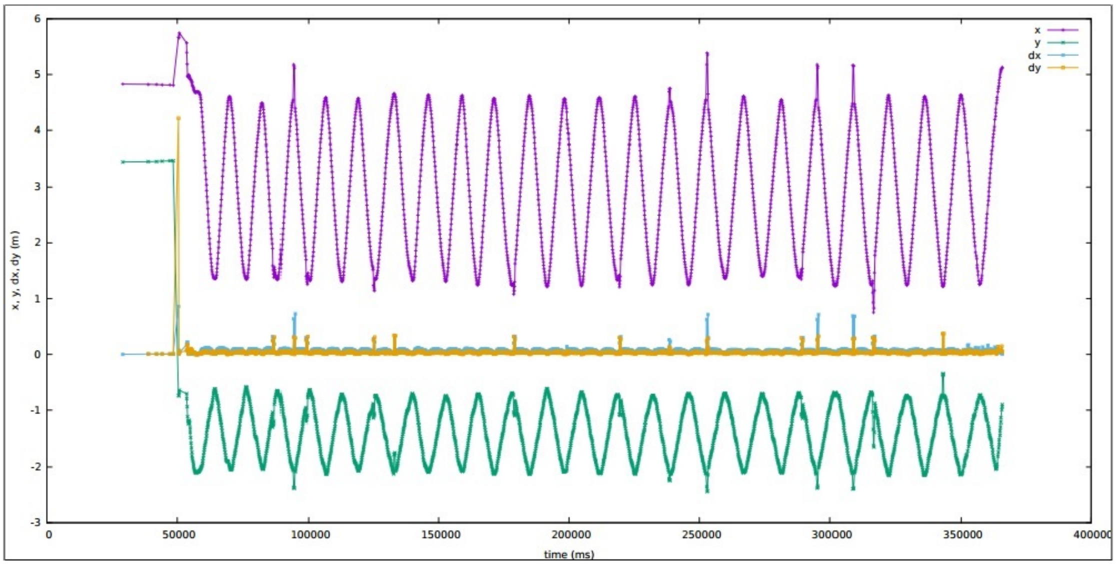


Figure 2.10: Showing separately X and Y trajectory path and including the difference between successive samples (dx,dy) of the person walked by putting on the mobile beacon for about 6 minutes

Chapter 3

High Sensitive and low noise capacitive sensor data acquisition

Range limited by noise and discretization

Major challenges in using capacitive sensors for long range localization is: after some distance like 1m, increasing distance results in very small change in capacitance. We need a capacitor sensor interface (data acquisition) design that overcomes this challenge. We are looking to develop an interface with low noise and exploit sensitivity at long distance (up to 2m) from the transducer (plate).

Interface designs for capacitive sensors includes converting the change in capacitinace to change in voltage/current and use fast and cheap (analog to digital converter) found in micro-controllers to digitize the voltage and use it as input to the Machine learning algorithm. The problem is at long distance the change is very small and the output voltage from the interface may not change at all, even if we have a very high resolution ADC, the change may not be detected.[2]

The authors[12] designed an interface that converts change in capacitance to change in frequency using RC based oscillators with 555 timers. The challenge is to measure the frequency accurately enough so that fractional changes are not missed and reduce the measuring time to increase speed and reduce power consumption (micro-controller stays in low power mode).

The front end design from the authors[12] is based on measuring frequency using a relaxation oscillator and attenuating the nosies using a base band digital filters. Then the frequency is measured using an MCU (ATmega328p micro-controller) and used by the localization algorithm to indicate/estimate the position of the person in tracking.

The effect of high-frequency noise on the output value from the sensors could be limited using low pass filters. This noises come from the environment (normal living home scenario) like switch and bulbs where the sensor operates. There is also "low-frequency drift noise" caused by leakage of static charges, temperature and humidity which can be reduced using high pass filters.[2]

In the next section the implementation of frequency measurement using ATmega328p is discussed.

3.1 Frequency Measurement

Measuring frequency of signals using micro-controller is common in applications.[19] Differences lies on the performance of the micro-controller used (in-terms of its operating clock frequency and structure of internal counters) and on the expected resolution of measurement or the acceptable error of the measured signal. Such parameters could change from application to application.

Having said that there are two techniques of measuring frequency each with its drawbacks and advantages.[19]

1.Measure the number of cycles in a fixed period of time:
As indicated in the fig 3.4, here the number of periods (cycles) are counted for a given amount of time. Frequency of the input signal in this case is given by[2]

$$FINPUT = N/T.$$

If time is longer this method gives us better accuracy but there is high need to keep measuring time shorter which results high error especially when measuring low frequency input signals.

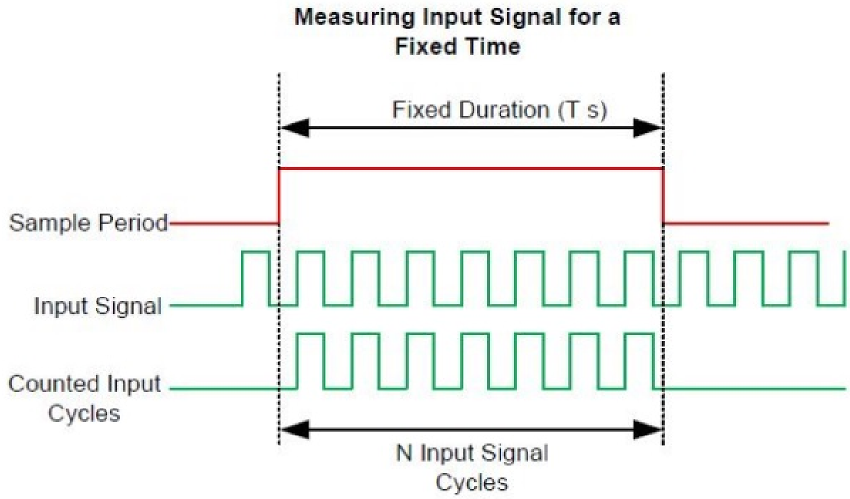


FIGURE 3.1: Measuring the Input Signal Cycles in a Fixed Duration

2.Measure the time of one cycle: In this case, shown on fig3.5, the number of clock cycles of a reference frequency is counted for one cycle of the input signal. The frequency for this case is given by[2]

$$FINPUT = \frac{FClock}{N}.$$

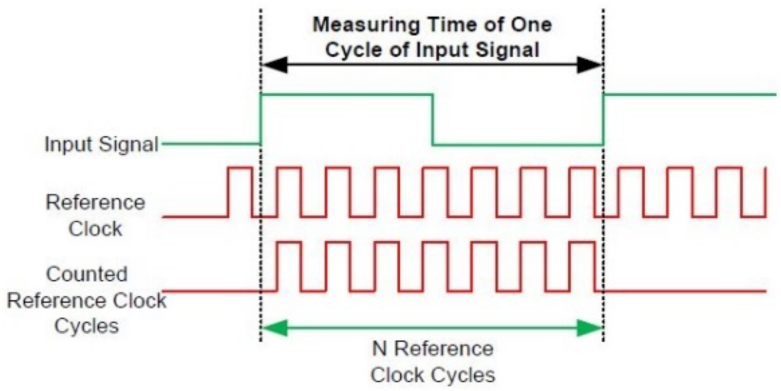


FIGURE 3.2: Measuring the Time Duration of One Input Signal Cycle

Contrary to the first one, this method is good for measuring low frequency but at high frequency measurement error increases.

Implementation method for ATmega328p micro-controller:

Limited by the capability and internal structure of ATmega328p, here another way is developed to measure the frequency of input signal. As it can be observed from fig3.7 a little modification of the first measuring method is done.

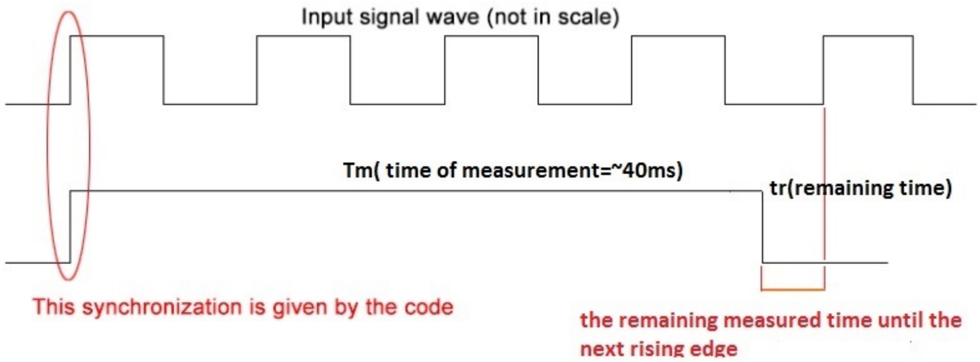


FIGURE 3.3: Implemented frequency measurement: counting full periods of the input signal using short measuring time

Still the signal will be measured for fixed amount of time using the clock frequency of the micro-controller. The goal is to measure full periods of the input signal, when the measuring time is finished instead of stopping measuring, the next rising edge of the input signal is waited on, which means though the counted input signal is not increased but our measuring time is extended to include the extra time for the next rising edge. The frequency of the input signal for this case is given by

$$FINPUT = \frac{N}{(Tm+Tr)}.$$

where N is the counted full periods of the input signal and Tm is the measuring time initially set, and Tr is the remaining time until the next rising edge of the input signal after measuring time (Tm) has elapsed.

By using this measuring technique accuracy is achieved. A fractional change on the input signal is captured.

To implement this approch on hardware requires two fast timers one clocked by high frequency clock signal (the operating clock frequency of the micro controller) and one clocked by the frequency of the input signal. Synchronizing the two timers is also important task of this implementation. Next section will describe the hardware implementation of this approach

Filtering: The output frequency coming from of the capacitive sensor has noise and shows fluctuations. To remove the noise median and averaging filters are used by the authors in [4]. In the experiment median filter with window size of 5 for short term acquisition is used while an averaging filter and offset-compensation is used for long-term acquisition.[2]

Normalizing Data: The output data from indoor positioning applications are either voltage or frequency and they need to be changed to distance. A reference output frequency from the sensor is taken as a base line value, which is measured when there is nobody inside the room. Further measurements are normalized to this baseline value. Normalization is done by subtracting measured values (person-sensor output frequency) from the reference value which is measured in empty room.[2]

Offset compensation: Variations of temperature and humidity and interferences coming from conductive objects near/in between/ sensor and the human body, affects the permitivity and conductivity of the sensors. Consequently the baseline value (the output frequency when there is nobody) shows a fluctuations or some variance. From many experiments it has been observed that this variance is around 1 percent.

The system should be able to differentiate between real signals and offsets at the same time updating the baseline value. A high-pass filter is implemented to compensate for offsets caused by environmental noises and the baseline value is updated when the measured signal is in a specified region for some time.[2]

3.2 Implementation

Here, we will discus about the implementation of the frequency measurement for ATmega328p micro-controller. The on going research described on [12] are done based on Arduino boards, enhancing this feature (frequency measuring) in-terms of accuracy and power optimization provides improvement on the analogue interface of the localization project.

3.2.1 ATmega328p microcontroller

As summarized on the table[20]

Features	ATmega328/P
Pin Count	28/32
Flash (Bytes)	32K
SRAM (Bytes)	2K
EEPROM (Bytes)	1K
General Purpose I/O Lines	23
SPI	2
TWI (I^2C)	1
USART	1
ADC	10-bit 15kSPS
ADC Channels	8
8-bit Timer/Counters	2
16-bit Timer/Counters	1

ATmega328p is "8-bit AVR RISC-based microcontroller consists 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts".[20]

The block digram is shown below

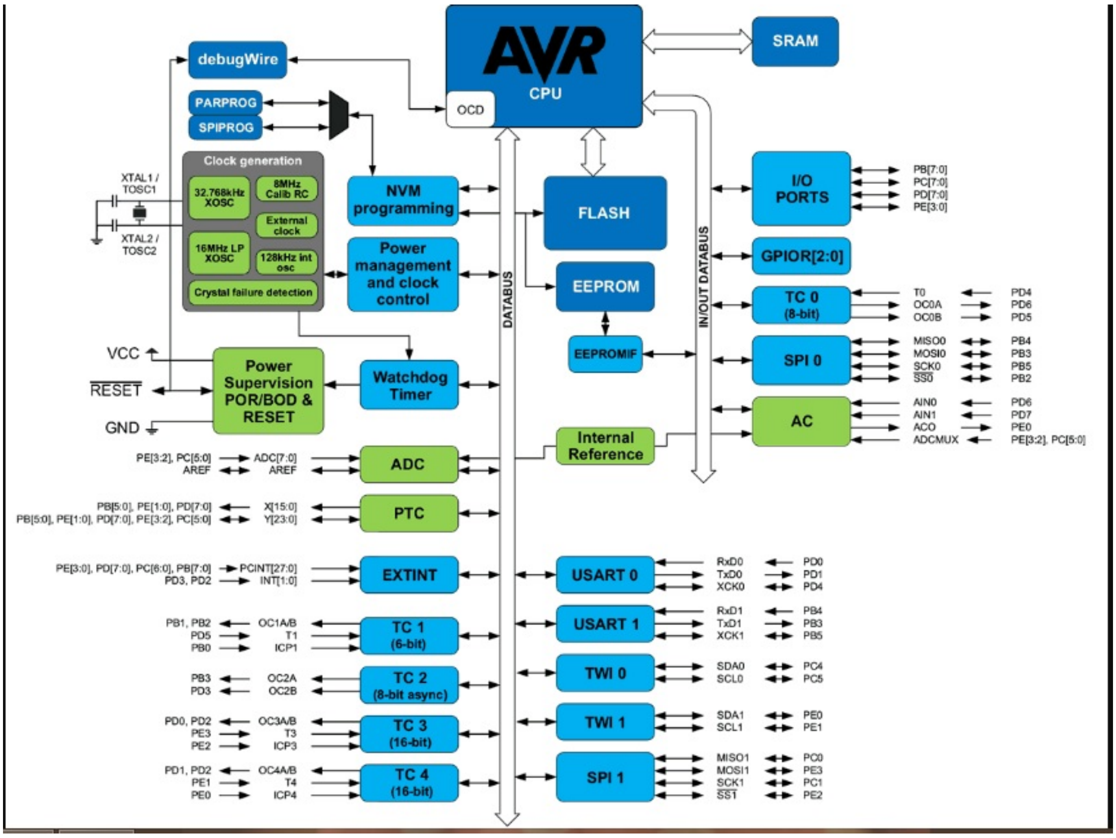


FIGURE 3.4: ATmega328p block diagram

3.2.2 Timers

[20] Looking how the timers works is helpful as most of the task of this project is implemented on th timers. ATmega328p has two 8 bit timers (called timer0 (T0) and timer2 (T2) and one 16bit timer (called timer1 (T1)). As example the block diagram of timer1 is shown on fig 3.8. Timer0 and timer1 can be configured to be

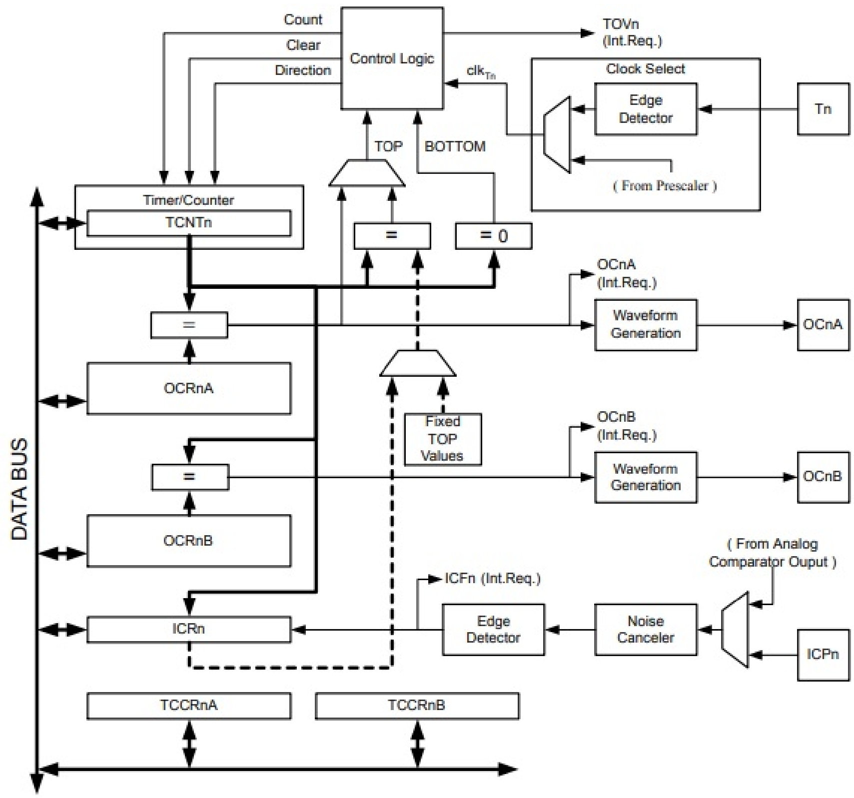


FIGURE 3.5: Block diagram of timer1

clocked by external clock signal. Based on the configuration (rising edge or falling edge). The timers increment/decrement count at the rising edge (default case) or falling edge of external signal.

3.2.3 Configuring the timers

First step is configuring the timers. To implement frequency meter described in the previous section we need timer0 to work on external clock which is our input signal to be measured (coming from the transducer after conversion to frequency by relaxation oscillator). The "Clock Select logic block" controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. If the clock select is not set the timer is inactive.[20]

Timer1 is the only timer working on 16bit and it is used to measure the "measuring-time" where highest resolution is required to measure the frequency of the input signal. So, timer1 will be working in default mode @16mHz which is the operating frequency of the micro controller.

Table Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}/1$ (No prescaling)
0	1	0	$clk_{I/O}/8$ (From prescaler)
0	1	1	$clk_{I/O}/64$ (From prescaler)
1	0	0	$clk_{I/O}/256$ (From prescaler)
1	0	1	$clk_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

As it can be seen from the table below, control registers are used to set the modes of Operation using control register called TCCR0B, and TCCR1B for timer0 and timer1 respectively.[20]

15.9.2 TCCR0B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

3.2.4 Synchronizing the timers:

Since the two timers are working on different frequency one coming from the outside one from the inside we need to synchronize them. We want the two timers to start equally both working on rising edges. This is also very important factor for the accuracy of the system:- measuring full periods requires synchronization.

From ATmega328p we can see many pin change hardware interrupt. For our frequency-meter implementation the external signal to be measured is also used to synchronize the two timers. This specific hardware pin (PD4/INT0/PINC20) has dual use: acting as external interrupt (T0) to be input clock signal for timer0 and acting as a pin-change interrupt to synchronize the two timers when starting the measurement.[20]

Again here some control registers are set to enable the pin-change interrupt.

PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

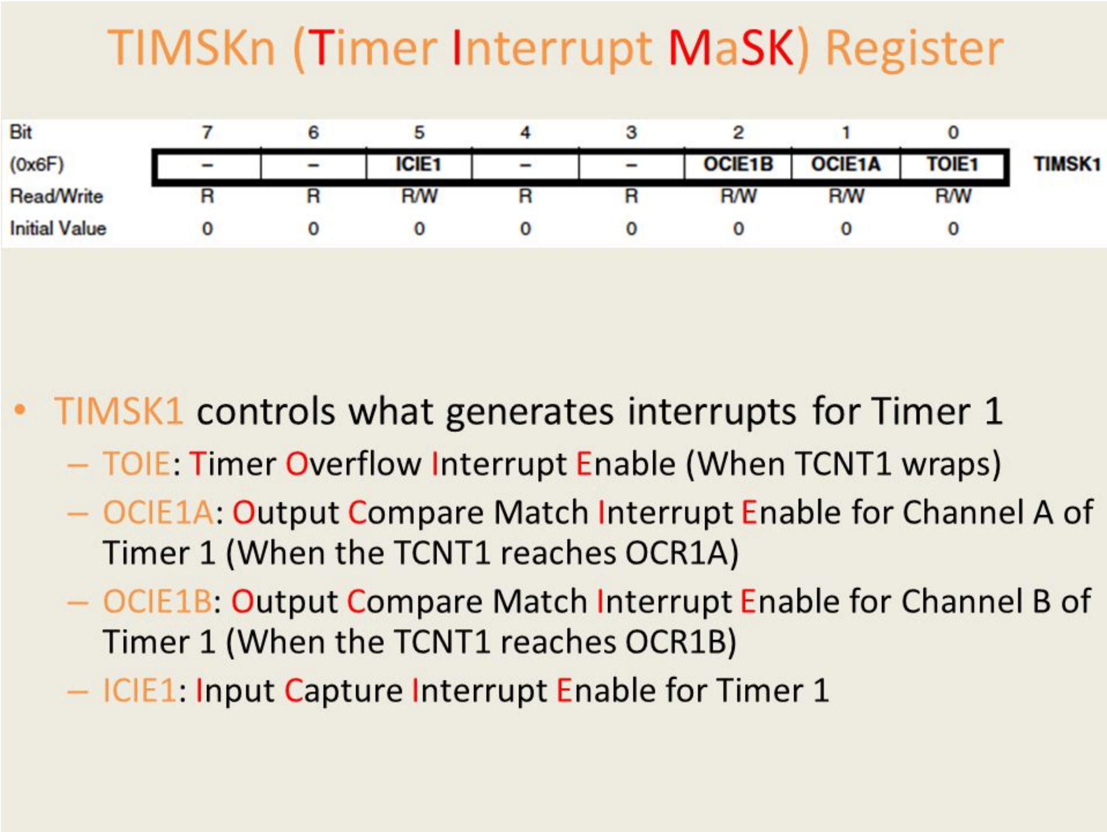
PCMSK1 – Pin Change Mask Register 1

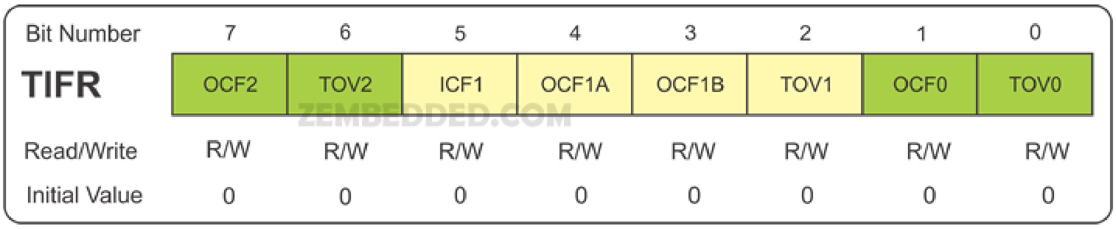
Bit	7	6	5	4	3	2	1	0	
(0x6C)	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Finally we need to set overflow interrupts for both of the timers. Firstly because our measuring time is a multiple of the overflows of timer1. One overflow takes 4ms, if measuring time is 40ms, 10 overflows are required. Secondly, timer0 is a small counter which overflows at 255 and if our input signal is fast, it will overflow many times before the measuring time finishes. By setting the control register related with overflow interrupt each overflow will be captured.[20]





3.2.5 Software

Here is a simple flow chart of the firmware and some codes that are necessary to handle the different interrupts, initializing and configuring the timers are described.

At the start, all configurations and working mode of timers are initialized. Then the firmware checks for a rising edge of the external signal on connected on PD4. At the rising edge the two timers (one for counting the external signal and the other for measuring time) are synchronized.

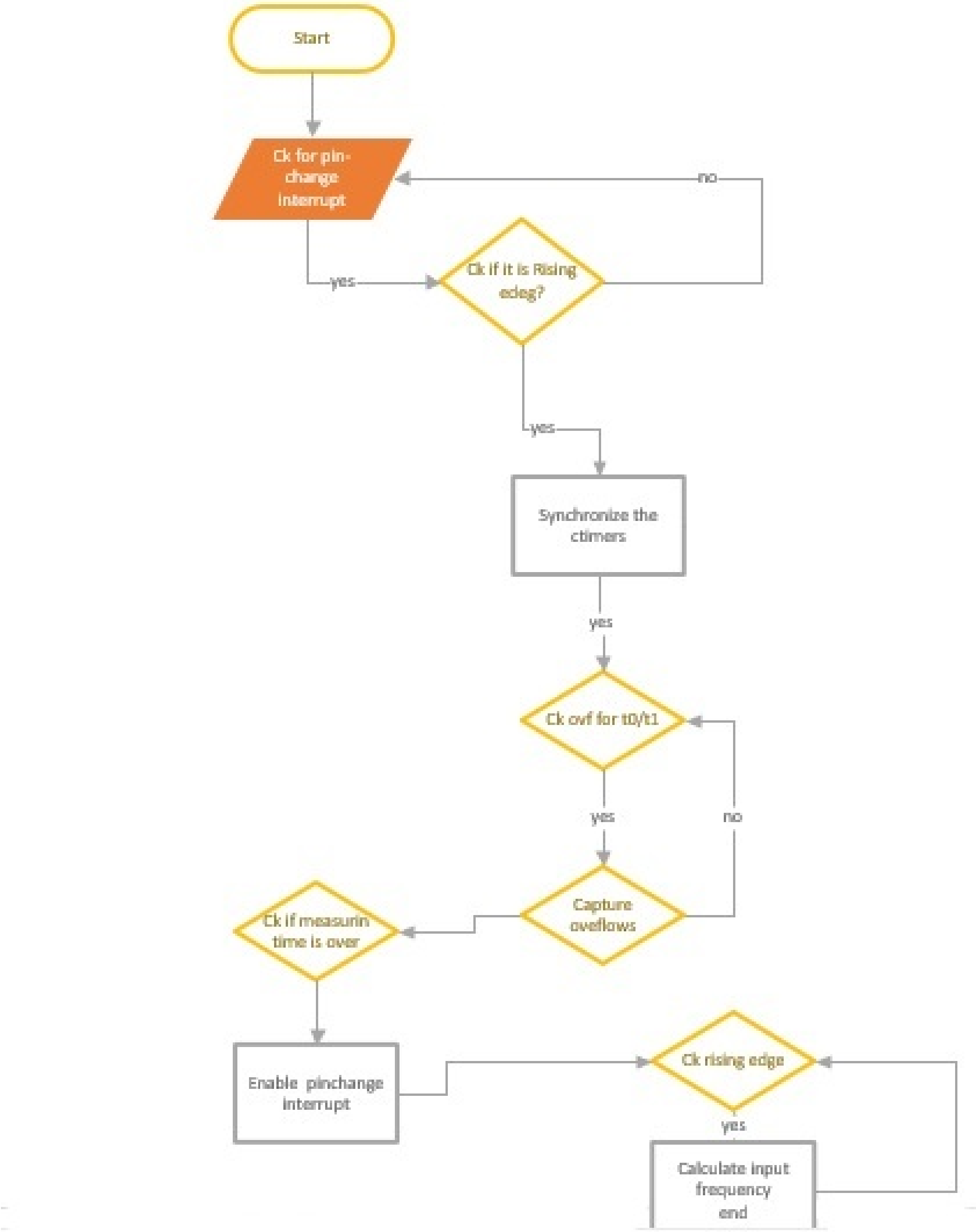


Figure 3.6: Flow chart of frequency measurement algorithm

Next after synchronization, the software checks for overflows of both timers and count the number of overflows in a variable. Our measuring time is based on a number of overflows of 16bit timer. For instance if we want to measure for 20ms Five overflows are counted as measuring time. There is also an overflow counter for timer0 (8bit), the one connected to the external signal. Because especially when measuring high frequency signal there are many overflows for a given measuring time.

Then the software checks if the measuring time has elapsed by comparing the counted 16bit timer overflow counter with initially set value.

After this, the software enables the pin-change interrupt to capture the last rising edge while the 16bit timer is counting the remaining time up to the last rising edge. Since we need to count full periods for better accuracy.

Finally the frequency of the input signal is computed by

$$Finput = \frac{N}{(TM + Tr)}.$$

Where N is the number of counted input full periods and Tm+Tr, makes the total measuring time.

3.2.6 Initialize and configure timers

In these section some code for initialization counter and interrupt handler are presented.

```
TCCR0A=0;                    // reset timer/counter0 control register A
TCCR0B=0;                    // reset timer/counter0 control register A
TCNT0=0;                     // counter value = 0

// timer1 setup / is used for frequency measurement gate time generation with 16MHZ/no prescaling
TCCR1A=0;
TCCR1B=0;

DDRD &= ~(1<<PORTD4); // Set pd4/pcint20 as pinchange interrupt
//PORTD |= (1<<PORTD4); /* Activate PULL UP resistor */
PCICR |= 0b00000100;    // turn on port d
PCMSK2 |= (1<<PCINT20);    // turn on pins PD4, PCINT2
```

initalization and cofiguration

The above code is for initialization of the timers, here it can be observed that few control registers that are responsible for configuring counters are accessed. By writing zero to this registers we are telling the counter to do nothing. At this point the counters are not responding to external/internal clock signals. The value of the counter register is also set with initial value=zero.

Pin-change interrupt

```
ISR(PCINT2_vect) // Port d, PCINt20
{
    statepin = PIND;

    if(((statepin & (0x10)) == 0x10) && (f_end_window==0)&& (f_start_window==0))
    {
        TCNT1=0;
        TCNT0=0; // reset couner1 and couner0
        count_ovf_t0=0;
        count_ovf_t1=0;
        TIMSK1 |=(1<<TOIE1 );        // enable Timer1 overflow Interrupt
        TIMSK0 |=(1<<TOIE0 );        // enable Timer0 overflow Interrupt
        // External clock source on t0 pin. Clock on rising edge.
        TCCR0B |= (1<<CS02) | (1<<CS01) | (1<<CS00);        //   start counting now
        //no prescaling on timer1

        TCCR1B |=  (0 << CS12) | (0<< CS11) | (1<< CS10);  //operates on the default mode
        PCMSK2  &= ~ (1<<PCINT20); //disable pin change interrupt
        f_start_window=1;
    }

    if(((statepin & (0x10)) == 0x10) && (f_end_window==1)&& (f_start_window==1))
    {
        f_main=1;
        TCCR1B = TCCR1B & ~7;                // Gate Off  / Counter T1 stopped
        TCCR0B = TCCR0B & ~7;                //  Counter t0 stopped
        input_periods=TCNT0;                 // collect the counted value
        extra_count=TCNT1;                  // take the reaming time waited on last rising edge
        PCMSK2  &= ~ (1<<PCINT20); //disable pin change interrupt
    }
}
```

pinchange intrrupt handler

Most of the task of the software part is done in the pin-change interrupter handler. This routine is responsible for synchronization of the two counter at starting point and to stop counters when measuring time is finished and values from the timers are extracted to be used for final computation.

The first section of the code is to reset the counter registers and overflow counter variables. It enables overflow interrupts of the two timers and set timer0 to work on external signal which is the input signal to be measured. Timer1 is set to work on internal operating clock frequency of the micro controller and lastly disables the pin-change interrupt since any change weather rising/falling edge on the pin cause the execution of this interrupt.

In the second section: measuring time is finished and at the rising edge the pin-change routine is called to stop the counters, and extract the counted value from TCNT1 and TCNT0 registers for the computation of input frequency.

Overflow handlers

```
// if Timer/Counter0 overflow flag
ISR(TIMER0_OVF_vect)
{
    count_ovf_t0++;               // count number of Counter0 overflows the external signal
}

// if Timer/Counter0 overflow flag
ISR(TIMER1_OVF_vect)
{
    count_ovf_t1++;

    if (count_ovf_t1 >= End_window) {
        f_end_window = 1;             // telling time measurement is ready flag
        // again enable pin change interrupt to capture the last rising edge of the input signal and complete the measurement of time
        PCMSK2 |= (1<<PCINT20);    // turn on pins for pin change interrupt
    }
}
```

overflow interrupt handler

Two separate overflow interrupt routine are presented for counting the number of overflows of the two timers. After 10 overflows are reached (which makes 40ms of measuring time) the pin-change interrupt is enabled to wait on the last rising edge.

3.2.7 Accuracy:

For a given time window the accuracy of the system can be calculated. In our case the time window is 40.96ms. The number of clock cycles counted in 40.96ms is equals to ($16MHz0.04096s = 655360$) clock cycles. If we assume there is a missing of one clock cycle at the starting point and one clock cycle at the end of the time window we have an $error = 2/655360 = 3.05x10^{-6}$. For shorter measuring time window the error increases since there is always a maximum of two clock cycles error. Using the above calculation for accuracy, if we take input signal with 100kHz frequency the $error = 100kHz * 3.05x10^{-6} = 0.305Hz$. With reduced time window, comparing to the previous systems the precision of this measuring system is higher and fractional changes on the input signal are captured.

3.3 Result

The experiment is conducted on arduino board using function generator as input signals at different selected frequencies (100kHz,90kHz,75kHz and 50kHz). First the out-put of the function generator is connected to the arduino board. The positive signal wire on the digital PIN4 (this pin is also called external interrupt INT0) of arduino board. The ground wire is connected to one of the ground pin of the arduino board.

The function generator is initially set to 50kHz, then incremented with 1Hz to see if the firmware can detect the small changes. The same procedure repeated for the selected frequencies (100kHz, 75kHz and 90kHz). The experimental setup is shown below on fig 3.7.

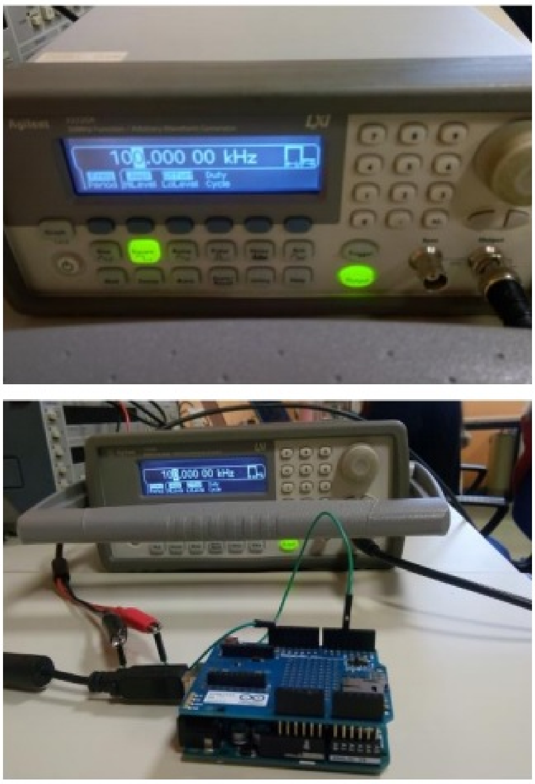


Figure 3.7: Experimental setup of testing the frequency meter

Precision is also checked by incrementing one and two on each selected frequencies (100kHz+1....and,100kHz+2) and the system was able to capture the changes, perhaps fractional changes. Measuring time is set to 40ms which is by far better from the previous system where measuring time was 1s. The result is also represented in the following plots for the selected frequencies on the fig 3.8 and 3.9.

75k	75k+1	75k+2	50k	50k+1	50k+2
75062.3749	75063.5234	75064.4374	50042.8476	50043.8437	50044.8359
75062.3749	75063.2968	75064.5546	50042.8476	50043.8437	50044.8359
75062.4921	75063.6406	75064.5546	50042.7773	50043.8437	50044.9101
75062.6093	75063.4062	75064.5546	50042.7773	50043.7695	50044.8359
75062.4921	75063.6406	75064.5546	50042.8476	50043.7695	50044.7617
75062.2656	75063.6406	75064.5546	50042.7773	50043.7695	50044.8359
75062.3749	75063.5234	75064.6718	50042.7773	50043.8437	50044.8359
75062.4921	75063.5234	75064.5546	50042.7773	50043.9179	50044.7617
75062.6093	75063.4062	75064.4374	50042.8476	50043.8437	50044.8359

FIGURE 3.8: Result obtained by incrementing a value of 1Hz difference at 75kHz and 50kHz

100k	100k+1	100k+2	90k	90k+1	90k+2
100084.6484	100085.7265	100086.4921	90069.5624	90070.2499	90071.6249
100084.6484	100085.7265	100086.4921	90069.2812	90070.3906	90071.4843
100084.6484	100085.7265	100086.4921	90069.5624	90070.2499	90071.4843
100084.6484	100085.414	100086.4921	90069.4218	90070.1093	90071.4843
100084.6484	100085.5703	100086.1796	90069.2812	90070.2499	90071.6249
100084.9609	100085.2578	100086.4921	90069.4218	90070.1093	90071.6249
100084.6484	100085.5703	100086.4921	90069.5624	90070.3906	90071.6249
100084.8046	100085.2578	100086.4921	90069.4218	90070.2499	90071.4843

FIGURE 3.9: Result obtained by incrementing a value of 1Hz difference at 100kHz and 90kHz

For better visualizations the above tabular results of the selected frequencies is plotted and presented in the following sections. Each color on the plot show that an increment of 1Hz from the base frequencies. The spikes with small amplitude are tolerable, they can be included in systematic errors and are not a problem for our measuring system. Another noticeable systematic error is that, there is always an increment to the base frequency. For example if the input frequency from function generator is 50kHz, the result from the micro-controller shows 50.042kHz. There is 42Hz extra clock period counted by the MCU. Since this value is constant for the measured frequency it can be avoided by simple offset mechanism.

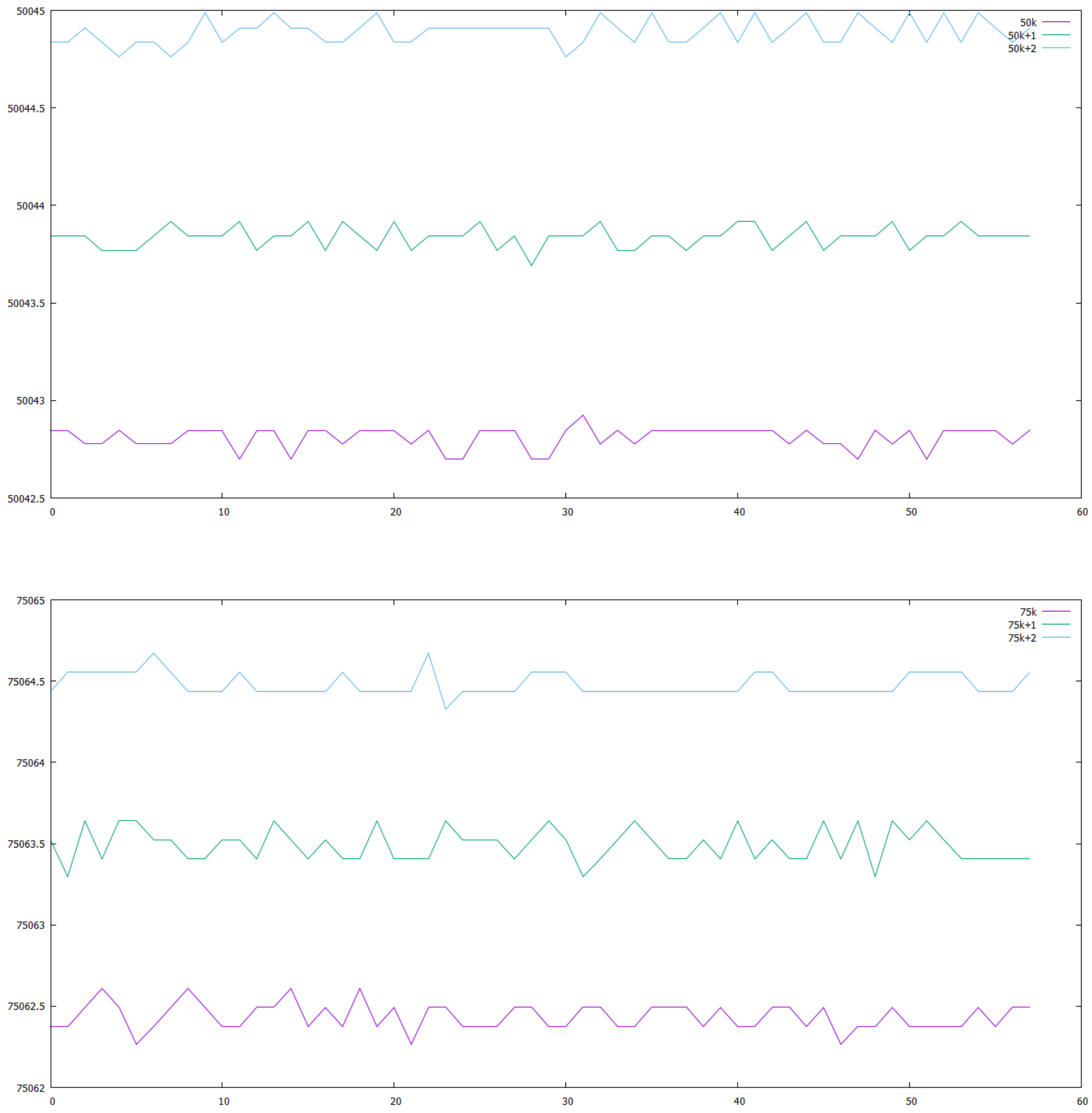


FIGURE 3.10: Result obtained by incrementing a value of 1Hz difference at,75k and 50kHz

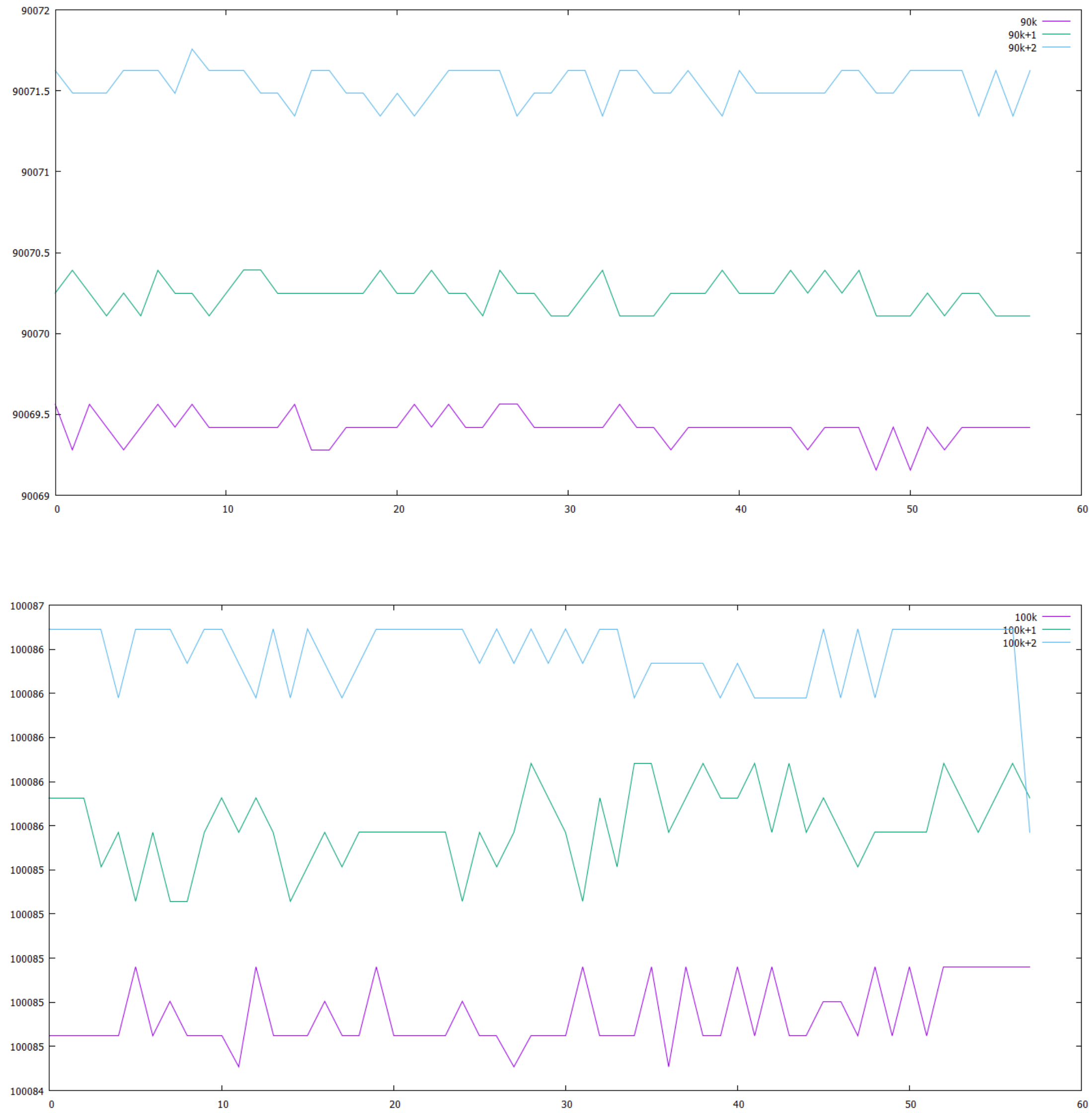


FIGURE 3.11: Result obtained by incrementing a value of 1Hz difference at 100k and 90kHz

After the validation of the firmware then it was also tested on the capacitive sensor circuit by using 8X8cm capacitor plate as shown in the following figure and started the measurement at 180cm with a step decrement of 40cm towards the sensor position. The system is showing better sensitivity from the previous one which was done on 16x16cm plate size.

Figure 3.12: Experimental setup for checking the response of the transducer (capacitive sensor)

A Matlab script is used to take samples at each positions. 20 samples per positions. The response time per sample is 1 second which means the person stands in one position for 20 second and makes a movement of an approximately 40cm towards the sensor. In total there are 6 positions and 120 samples. Some plots of the response of the sensor are shown on the fig3.11-3.13 below. They show good sensitivity even at the end of the measurement range and a relatively low level of noise.

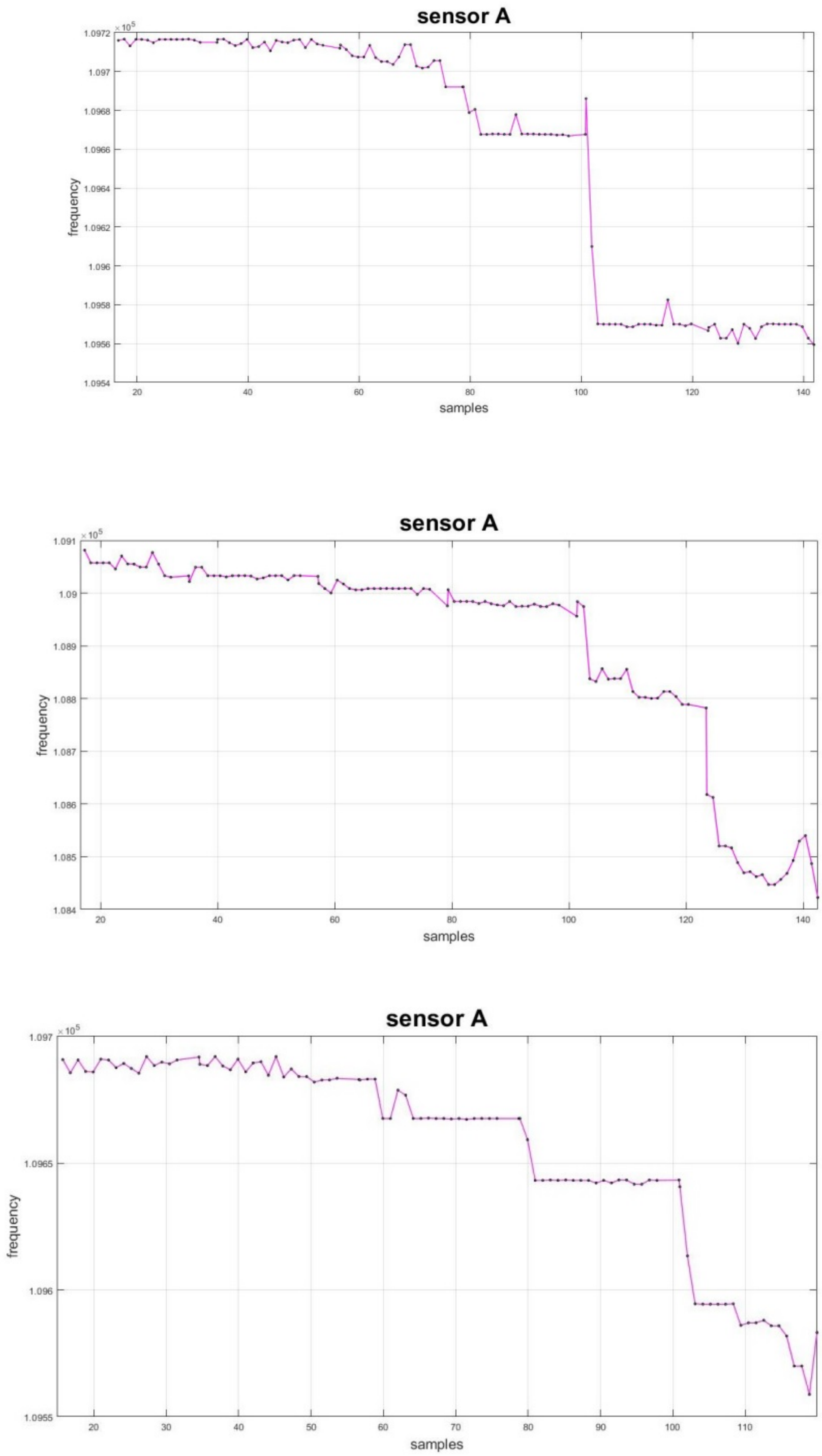


FIGURE 3.13: Multiple responses of the capacitive sensor: the farthest measurement being at 180cm and the closest at 50 cm away from the plate. 20 samples represent one position

Chapter 4

Conclusion

To optimize tag-less long distance human localization project this thesis contribute two aspects of optimization: The first is characterizing ultrasonic based indoor navigation system which is an off-shelf product acquired from Marvel-mind robotics.

Ultrasonic sensors integrated with RF (radio frequency) communication interface (for the purpose of synchronization) are used for indoor positioning systems (IPS). These sensors are preferable from other same kind application systems: comparatively they have low-cost of infrastructure, high accuracy and low power consumption.

These commercial sensors were tested to see if the specified accuracy and speed of location update is correct and also see if they are reliable to be used as many label data inputs for localization algorithm.

From the experiment the maximum error for 16 fixed position test (20samples per position were taken) is about 10-12cm. The maximum error between two successive sample points is 20cm. The system is able to track a person with a resolution of few centimeters.

Most of the experiment result shows that the data are in the expected range but some spikes were observed. This could be because the experiment is conducted in a lab environment, where there are many instruments that could interfere with the ultrasonic sensors. The way the sensors are installed is also an important factor in reducing errors (a line of sight and a specific amount of distance between adjacent beacons is required). Moreover the accuracy of the sensor is affected by temperature and humidity variations. If the sensor is tested in controlled enviroment by avoiding the listed limitation, for sure better accuracy can be obtained.

Overall, with those limitation the indoor navigation system proved to be used as reference system for the training and testing of the localization algorithm.

Secondly this thesis contributed for the optimization of the analogue front end interface. Here the focus is optimizing the frequency meter by increasing accuracy and reducing measuring time which in-turn reduces power consumption.

The output frequency from the relaxation oscillator is measured by using ATmega328p micro-controller. The developed firmware was tested and resulted accurate frequency meter with 40ms measuring time. Compared to the previous measuring method which took one second, in this system higher speed and accuracy is achieved.

ATmega328p has limitations in-terms of speed (operates @16mHz) and internal structure (only two timers are available to be used). Better results and simplicity in designing could be achieved by working on other micro controller with higher speed and better capability of internal structures.

Appendix A

frequency-meter code

```

#include <avr/io.h>
#include <stdlib.h>
#include <math.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#define F_CPU  16000000UL
#define BAUDRATE 9600
#define BAUD_PRESCALLER (((F_CPU / (BAUDRATE * 16UL))) - 1)

/////// These variables for sending over XBEE ////

#define highWord(w) ((w) >> 16)
#define lowWord(w) ((w) & 0xffff)
#define highByte(b) ((b) >> 8)
#define lowByte(b) ((b) & 0xff)
uint8_t payload[4] ;
uint16_t addr = 0xABCD;
uint16_t hiword_frq, loword_frq;
long to_rmv_float;

//////////////////////////////////////////////////
unsigned long measuring_window;
float true_measuring_window;
unsigned long extra_count;
```

```
unsigned long total_input_periods;
unsigned long input_periods;
float frq;
volatile unsigned char f_start_window;
volatile unsigned char f_end_window;
volatile unsigned char f_main;
volatile unsigned char count_ovf_t0;
volatile unsigned char count_ovf_t1;
int End_window;
char buffer1[20];
char statepin;
void start(int EP);
void USART_init(void);
unsigned char USART_receive(void);
void USART_send( unsigned char data);
void USART_putstring( char* StringPtr);
long intToStr(long x, char str[], int d);
void ftoa(float n, char *res, int afterpoint);
void reverse(char *str, int len);
#if defined (__AVR_Ameasuring_windowega168__) || defined (
    __AVR_Ameasuring_windowega48__) || defined (
    __AVR_Ameasuring_windowega88__) || defined (
    __AVR_Ameasuring_windowega328P__) || (
    __AVR_Ameasuring_windowega1280__)
#endif

// if Timer/Counter0 overflow flag is up
ISR(TIMER0_OVF_vect)
{
count_ovf_t0++;                // count number of Counter0 overflows
    the external signal
}

// if Timer/Counter0 overflow flag
ISR(TIMER1_OVF_vect)
{
count_ovf_t1++;
if (count_ovf_t1 >= End_window) {
```

```
f_end_window = 1;               // telling time measurement is ready
    flag
// again enable pin change interrupt to capture the last rising edge
    of the input signal and complete the measurement of time
PCMSK2 |= (1<<PCINT20);     // turn on pins for pin change interrupt
}

}
//pin change interrupt routine
ISR(PCINT2_vect) // Port d, PCINt20
{
statepin = PIND;
if(((statepin & (0x10)) == 0x10) && (f_end_window==0)&& (
    f_start_window==0))
{
TCNT1=0;
TCNT0=0; // reset couner1 and couner0
count_ovf_t0=0;
count_ovf_t1=0;
TIMSK1 |=(1<<TOIE1 );         // enable Timer1 overflow Interrupt
TIMSK0 |=(1<<TOIE0 );         // enable Timer0 overflow Interrupt
// External clock source on t0 pin. Clock on rising edge.
TCCR0B |= (1<<CS02) | (1<<CS01) | (1<<CS00);          //   start
    counting now
//no prescaling on timer1
TCCR1B |=  (0 << CS12) | (0<< CS11) | (1<< CS10);
PCMSK2  &= ~ (1<<PCINT20); //disable pin change interrupt
f_start_window=1;
}

if(((statepin & (0x10)) == 0x10) && (f_end_window==1)&& (
    f_start_window==1))
{
f_main=1;
TCCR1B = TCCR1B & ~7;            // Gate Off  / Counter T1 stopped
TCCR0B = TCCR0B & ~7;            //  Counter t0 stopped
input_periods=TCNT0;
extra_count=TCNT1;
```

```
measuring_window = count_ovf_t1*65536 + extra_count;
 true_measuring_window = ((float)measuring_window*0.004096)/65536;
total_input_periods = count_ovf_t0*256 + input_periods;
//f_end_window=0;
f_start_window=0;
f_main=0;
PCMSK2  &= ~ (1<<PCINT20); //disable pin change interrupt
}
}

int main(void)
{
cli();
End_window=10;
TCCR0A=0;                     // reset timer/counter0 control register A
TCCR0B=0;                     // reset timer/counter0 control register A
TCNT0=0;                    // counter value = 0

// timer1 setup / is used for frequency measurement gate time
    generation with 16MHZ/no prescaling
TCCR1A=0;
TCCR1B=0;
f_end_window=0;
f_start_window=0;
//f_main=0;
//pinchange interrupt
DDRD &= ~(1<<PORTD4); // Set pd4/pcint20 as pinchange interrupt
//PORTD |= (1<<PORTD4); /* Activate PULL UP resistor */
PCICR |= 0b00000100;    // turn on port d
PCMSK2 |= (1<<PCINT20);    // turn on pins PD4, PCINT2
sei();
while (1)
{

while(f_end_window==1){
if (f_main==0) {
float true_measuring_window = ((float)measuring_window*0.004096)
    /65536;
```

```
frq= ((float)total_input_periods)/(true_measuring_window);    //
    measuring time(measuring_window) is not 1 sec but it is arrnd 40ms
// changing to strings for usart communication
//ftoa (frq, buffer1, 4);
USART_init();
//USART_putstring(buffer1);
//USART_putstring("\n");
// }

/////// This section is for XBEE communication ////////
to_rmv_float = frq*100;
hiword_frq = highWord(to_rmv_float);
loword_frq = lowWord(to_rmv_float);

payload[0] = highByte(hiword_frq);
payload[1] = lowByte(hiword_frq);
payload[2] = highByte(loword_frq);
payload[3] = lowByte(loword_frq);

XBee_TX_Request(addr, payload, 4);
}
/////// End of XBEE communication section /////////
_delay_ms(1000);
PCMSK2 |= (1<<PCINT20);    // turn on pins PD4, PCINT2
f_end_window=0;

}
}
}

void USART_init(void){

UBRR0H = (uint8_t)(BAUD_PRESCALLER>>8);
UBRR0L = (uint8_t)(BAUD_PRESCALLER);
UCSR0B = (1<<RXEN0)|(1<<TXEN0);
UCSR0C = (3<<UCSZ00);
}

```

```
unsigned char USART_receive(void){

while(!(UCSR0A & (1<<RXC0)));
return UDR0;

}

void USART_send( unsigned char data){

while(!(UCSR0A & (1<<UDRE0)));
UDR0 = data;

}

void USART_putstring( char* StringPtr){

while(*StringPtr != 0x00){
USART_send(*StringPtr);
StringPtr++;}

}

void reverse(char *str, int len)
{
int i=0, j=len-1, temp;
while (i<j)
{
temp = str[i];
str[i] = str[j];
str[j] = temp;
i++; j--;
}
}
// Converts a given integer x to string str[].
long intToStr(long x, char str[], int d)
{
int i = 0;
while (x)
```

```
{
str[i++] = (x%10) + '0';
x = x/10;
}
// If number of digits required is more, then
// add 0s at the beginning
while (i < d)
str[i++] = '0';
reverse(str, i);
str[i] = '\0';
return i;
}

// Converts a floating point number to string.
void ftoa(float n, char *res, int afterpoint)
{
// Extract integer part
long ipart = (long)n;

// Extract floating part
float fpart = n - (float)ipart;

// convert integer part to string
long i = intToStr(ipart, res, 0);

// check for display option after point
if (afterpoint != 0)
{
res[i] = '.';  // add dot
// Get the value of fraction part upto given no.
// of points after dot. The third parameter is needed
// to handle cases like 233.007
fpart = fpart * pow(10, afterpoint);
intToStr((long)fpart, res + i + 1, afterpoint);
}
```

LISTING A.1: frequency-measuring

Bibliography

[1] Tero Kivimäki, Timo Vuorela, Pekka Peltola, and Jukka Vanhala. A review on device-free passive indoor positioning methods. *International Journal of Smart Home*, 8(1):71–94, 2014.

[2] Pooya Poolad. Optimization of capacitive sensor front-end for indoor human localization and identification. 2017.

[3] P. Peltola Kivimki, T. Vuorela and J. Vanhala. a review on device-free passive indoor positioning methods, international journal of smart home, vol. 8, no. 1,. pages pp. 7194, 2014.

[4] Osama Bin Tariq Alireza Ramezani Akhmareh, Mihai Teodor Lazarescu and Luciano Lavagno. A tagless indoor localization system based on capacitive sensing technology.

[5] Raphael Wimmer, Matthias Kranz, Sebastian Boring, and Albrecht Schmidt. A capacitive sensing toolkit for pervasive activity detection and recognition. In *Pervasive Computing and Communications, 2007. PerCom'07. Fifth Annual IEEE International Conference on*, pages 171–180. IEEE, 2007.

[6] Tobias Grosse-Puppendahl, Christian Holz, Gabe Cohn, Raphael Wimmer, Oskar Bechtold, Steve Hodges, Matthew S Reynolds, and Joshua 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*, pages 3293–3315. ACM, 2017.

[7] https://en.wikipedia.org/wiki/capacitance.

[8] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 32–43, New York, NY, USA, 2000. ACM. ISBN 1-58113-197-6. doi: 10.1145/345910.345917. URL http://doi.acm.org/10.1145/345910.345917.

[9] Mark Hudson Beale, Martin T Hagan, and Howard B Demuth. Neural network toolbox user's guide. *The Mathworks Inc*, 1992.

[10] Nils J. Nilsson. Introduction to machine learning. https://ai.stanford.edu/~nilsson/mlbook.html, 1998. [Online; accessed 4-July-2017].

[11] Andrew Ng. Machine learning. https://www.coursera.org/learn/machine-learning. [Online; accessed 4-July-2017].

[12] Osama Bin Tariq, Mihai Teodor Lazarescu, Javed Iqbal, and Luciano Lavagno. Performance of machine learning classifiers for indoor person localization with capacitive sensors. *IEEE ACCESS*.

[13] $http : //education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html.$

[14] 2 Guangjie Han 1 Chunsheng Zhu 3 Jian Li, 1 and Guiqing. An indoor ultrasonic positioning system based on toa for internet of things. "2016".

[15] Jose Carlos Segura Carlos Medina and Angel De la Torre. Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy. 2013.

[16] Chankil Lee" "Faheem Ijaz. Indoor positioning: A review of indoor ultrasonic positioning systems.

[17] Hiroyuki MORIKAWA Yasuhiro FUKUJU, Masateru MINAMI and Tomonori AOYAMA. An autonomous indoor positioning system in ubiquitous computing environment.

[18] www.marvelmind.com. $Marvelmindindoornavigationsystemoperatingmanual, v2018_0 1_1 1.$

[19] Dave Van Ess. Measuring frequency.

[20] Atmel. $http : //ww1.microchip.com/downloads/en/devicedoc/atmel - 42735 - 8 - bit - avr - microcontroller - atmega328 - 328p_datasheet.pdf$.