UNIVERSITY OF NAIROBI

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

ROOM OCCUPANCY MONITORING AND COUNTING DEVICE

Project Index: 76

By

Okoth Francis Ogollah

F17/36655/2010

Supervisor: Prof. J Mwangi Mbuthia

Examiner: Prof. Maurice Mang'oli

Project report of the final year project towards partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering of the University of Nairobi

Submitted on: April 28, 2015

DECLARATION OF CERTIFICATION

NAME: OKOTH FRANCIS OGOLLAH

REGISTRATION NUMBER: F17/36655/2010

COLLEGE: Architecture and Engineering

FACULTY/SCHOOL/INSTITUTE: Engineering

DEPARTMENT: Electrical and Information Engineering

COURSE NAME: Bachelor of Science in Electrical and Electronic Engineering

TITLE OF WORK: Room occupancy and counting Device

1) I understand what plagiarism is and I am aware of the University Policy in this regard.

2) I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of degree or publication. Where other people's work or my work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements

3) I have not sought or used the services of any professional agencies to produce this work

4) I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work

5) I understand that any false claim in respect of this work shall result in disciplinary action, in accordance with the University anti-plagiarism policy

This is my original work and has not been presented for a degree award in this or any other university.

……………………………………………………………

OKOTH FRANCIS OGOLLAH

F17/36655/2010

This report has been submitted to the Department of Electrical and Information Engineering, University of Nairobi with my approval as supervisor:

………………………………………………..

Prof. J. Mwangi Mbuthia

Date:25th April 2015

TABLE OF CONTENT

Contents

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEGEMENT

I welcome this opportunity to express our heartfelt gratitude and regards to our Professor J . Mwangi Mbuthia for his unconditional guidance. He always bestowed parental care upon us and evinced keen interest in solving our problems. I consider myself fortunate to have worked under his supervision.Without his co-operation, the extensive work involved in compiling background information and preparing the paper for publication would not be possible.

I would also like to thank my mum for her encouragement and prayers during the time I was undertaking this project. Her enormous support gave me the energy and ability to complete this task.

May God Almighty bless you in their endeavors for their guidance and support that enabled me complete this work successfully

Dedication

This project work is dedicated to the final year Electrical Engineering class of 2015 at the University of Nairobi for all the best moments we had together.

ABSTRACT

Monitoring and counting of people is very popular in many commercial and non-commercial scenarios. The number of people entering and leaving shops, the occupancy of office buildings, companies, museums or libraries provides useful information to shop merchants and marketers, security officials or the human resource officials the statistics of occupancy. This project develops a monitoring and counting device which uses suitable algorithms and fuzzy rules to implement on PIR Am412 sensors. For evaluation, a prototyping processor board and relatively low-cost PIR sensors were used to build two sensor nodes from commercially available components. A feature of this implementation is that each sensor node is able to count, on-system and in real-time, the number of passers-by going in each direction, while requiring relatively little compute power. The sensors network is described to monitor the count the number of people entering and leaving a room then update these data on a screen. To enable the configuration of the sensor nodes to specific installation scenarios, and for the purpose of performance analysis, a validation framework is presented.

Motion detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object. Motion detection can be achieved by both mechanical and electronic methods. The mechanical motion detection is in the form of a switch or trigger while the electrical motion detection is identified are optical detection and acoustical detection. Infrared light or laser technology may be used for optical detection. Motion detection devices like PIR motion detectors have a sensor that detects a disturbance in the infrared spectrum, such as a person or an animal. Once detected, an electronic signal can activate a system or a camera that can capture an image or video of the movement.

This project focuses on using motion detectors to develop fuzzy rules and suitable algorithms which are then applied to build a room occupancy and counting device. The overall project is divided into two parts namely the hardware development and the software programming. In the hardware structure, all the electronic components are connected via circuit design i.e. the motion detectors/sensors are the input components while the screen (display) is the output component as it is controlled by a controller circuit (microcontroller). The second part is based on software development (programming) to operate the hardware structure.

Programs based on fuzzy rules and suitable algorithms are used on the microcontroller which assembles the code to get the binary file then loads it into an external memory of the hardware structure via the serial communication. This process of transferring and executing the program is done using HyperTerminal's communication software to the microcontroller serial port. The best room occupancy and counting device is obtained by connecting more motion detectors or sensors to the microcontroller's output port. Based on the conditional statements in the algorithms, programs using software like MATLAB command and COOCOX COIDE codes that enables the microcontroller to control the counting process as the motion sensors detect motion.

CHAPTER 1 INRODUCTION

1.1 General Background

A motion detector is defined as a device that contains a motion sensor which is either integrated with or connected to the other devices that alerts the user of the presence of motion. A motion can be detected by measuring the change in speed or vector of an object in the field of view. Thus, it can be achieved either by mechanical devices that physically interacts with the field or by an electronic device that measures the change in the given environment. In mechanical devices, if a moving objects steps into the field of view, a simple sound like device will alert the user. The electronic motion detectors contains a motion sensor that detects any physical movement in a given area then transforms the detection into electrical signal. The electrical signal can then be connected to a control logic device where based on conditional reasoning and algorithm enhances the counting as motion is detected

Motion can be detected by detected by different methods i.e. sound (acoustic sensors), opacity (optical, infrared sensors and video image processing sensors), geomagnetism (magnetic sensors, magnetometers), reflection of transmitted energy (infrared laser radar, ultrasonic sensors and microwave radar sensors), electromagnetic induction (inductive-loop detectors) and vibration (triboelectric, seismic, and inertia-switch sensors).

Motion is always observed and measured relative to a frame of reference. A body is still moving relative to infinitely many others fames although the body which is motionless relative to a given reference fame. There are many approaches for motion detection in a continuous video stream. All of them are based on comparing of the current video frame with one from the previous frames or with something that called as background. In this paper, there are four approaches are used and comparison is made to find out a best detector for an effective motion detection.

There are two types of motion sensors namely the area and local sensors. The area sensors i.e. passive infrared, proximity(RF fields), microwave, radar, ultrasonic, vibration etc. do not send any signal out but they merely receive signals like changes in temperature or light intensity. They detect the heat energy radiated or emitted by an object like a human body moving across a field of view of the heat sensor. The logic of passive infrared sensor is that it must detect the significant change of the normal level of heat within the field of view so that its circuit control must be able to determine what is normal and close its switch when the normal field changes. It must also be able to 'tolerate' slow changes with the field so that the gradual changes like sunlight changes throughout the day so as not to causes false detection.

The local sensors i.e. active infrared, laser beam, visible light beam, contact, tilt etc. uses a light to scan a defined area. It sends out waves of energy and receive waves reflected back from the object so any disturbance in the reflected waves caused by a moving object will trigger for motion detection. The active infrared systems has two pieces of elements namely infrared transmitter and receiver. It has a 0.9525 cm infrared beam between the transmitter that is placed on one side of the trail and the receiver that is placed on the other side. The separation distance can be as much as 150 feet (45.72m).

The appropriate fuzzy reasoning (i.e. the conditional statements in the form of IF …THEN..) and algorithm (suitable method followed while solving a problem) are required since the motion sensors will sense on any motion i.e. flying insect, pets, human beings. This is because all objects on the earth emit light in accordance with their temperature and surface characteristics. Naturally, light (infrared radiation) is also emitted from our bodies. (This radiation is emitted from the body surface, and is centered around a wave-length of 10μm.) When a person enters the detection area of the sensor, the amount of infrared radiation incident on the sensor varies by the amount corresponding to the difference in temperature between the body surface and background. Thus, the appropriate distinction should be done so that the counting device only counts if a certain threshold of the signal intensity is crossed. The sensors are also to be arranged in a certain specific way hence the rules and algorithm will help in detecting movement into the building and movement out of the building or if the person is just idling on the entrance.

1.2 Objectives

1.2.1 Overall Objective

The main objective of this project is to use motion detectors i.e. Passive Infrared to develop fuzzy rules and a suitable algorithm and apply them to build a room occupancy and counting device.

1.2.2 Specific Objective

The following areas of knowledge needs to be understood properly to help in accomplishing the overall objectives of this project

- Understanding the principal of operation of motion detectors
- Understanding the concept of fuzzy logic
- Coming up with a suitable algorithm for counting
- Applying above concept in building the room occupancy monitoring and counting devise

1.3 Project Justification

If there is any technological advancement that would certainly make life easy and convenient, then sensors would be the answer . They have shown significant in decreasing human work load especially in industries. They are majorly used in digital control engineering for monitoring measurements changes on the feedback path.

1.3 Project Scope

This project aims at building a room occupancy and counting device using the passive infrared sensors as motion detectors. The sensors are used to develop human like reasoning principles that are organized in a certain problem solving method. The whole project is developed on two schemes namely the electrical hardware scheme and the software programming scheme.

The passive sensors are to be used to alert the microcontroller in case there is any motion within the specified area of view. The control program will then determine whether the counter should be incremented or decremented depending on the direction of the detected motion. The electrical component will only have two systems namely the passive infrared circuit which detects the motion and the microcontroller circuit which controls the whole operation of counting the number of people in the room. Its software programming module is based on the MATLAB command software and microcontroller instruction set to be used. The program codes developed for the device must operate the electrical module.

Report Organization

The subsequent chapters provides a broad understanding of fuzzy concept, counting algorithm and building of device.

The Second Chapter outlines the literature review . It gives the details of motion detectors/sensors history in counting people. The principle of concept that have been previously used by different sensors to achieve the counting, there advantages and disadvantages have been discussed. It also gives the sensor settled upon and why.

The third Chapter focus on the Methodology of the design. It gives the design algorithm used i.e the arrangement of the sensors and the program flow of events. It explains the principle of operation of the used algorithm and also the fuzzy concept.

The forth Chapter provides the process of Implementation. It gives the technical aspects involved in lying out the project

The fifth Chapter discusses the Results and Evaluation of the project. The pictorial results obtained from the demonstrations are show and interpreted.

The sixth Chapter gives the Conclusion and Recommendation of the project. It give the final comment based on the project performance and its future work.

Finally the outlines of the Appendix of the project. It gives the financial budget for development of the prototype. It gives the time budget during the development of the project. The program codes for MATLAB command and the COOCOX COIDE.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

Motion sensors are used in wide variety of applications and as a results, many different types of motion sensors are available i.e. infrared sensor. The infrared sensors have basically two forms namely the **active(Area Reflective Sensor)** and **passive infrared sensors**. The examples of Area Sensors Include passive infrared, proximity (RF Fields), Microwave, Radar, Ultrasonic, Vibration and Video sensors. The examples of Local sensors includes active infrared, Visible light beams, Laser Beams, Contact, Tilt, Strain and stress sensors. Nowadays most motion sensors are wireless. The wireless sensors are very easy to set up. Since do not require drilling and they also communicate with the other security system components wirelessly.

2.2Active(Area Reflective) Sensor

An active infrared detector emits infrared rays from an LED then using the reflection of those rays the sensor measures the distance to the person(object) and detects whether or not it exists within a specific distance.It includes the source radiation and an infrared sensor that is sensitive to interruption in the radiation sensed from the source. The detector provides a path of radiation from the source of the sensor to the interruption.

advantages

1. Fast speed of response of the relatively large sensor that permits simpler optical system design especially for wide fields of view.

2.It is insensitive to mechanical and acoustic noise compared to the passive infrared sensor. It also have low production cost.

3. It is also possible for detection within specific ranges i.e it is possible to select the detection distance.

2.3 Infrared Sensor Arrays

The IR arrays combine a matrix of IR sensors to form array detectors. As the name suggests the sensor signals are provided as a matrix, where each element of the matrix corresponds to one IR sensor. This system uses infrared (IR) arrays and pattern recognition algorithms. The pattern recognition algorithms are able to detect people moving across the sensor's view at a claimed of very high accuracy. It holds true even if two pedestrian's paths cross, or people walk in parallel. IR arrays provide a cost-effective solution and also operate without any ambient light source. IR arrays are widely used in commercial systems.

2.4 Passive Infrared (PIR):

The passive infrared or electrolytic or IP motion sensor (PIR) are basically made of a electrolytic sensor that can detects the level of infrared radiation. It detects (infrared energy) heat energy radiated or emitted by an object like a body of a person moving across a field of view of a heat sensor of the motion detection system. It generally use an optical system collection and multiple sensing elements of alternating polarity to create a detection pattern in the volume of interest. The PIR detectors employs a group of radiation sensors coupled through amplifiers to a logic circuit. The detection system has an electrical circuit imperatively coupled to the heat sensor for producing a detection signal in response to the heat sensor detecting a change of temperature caused by the body heat of a person entering the detection pattern.

When your system is armed, your motion sensors are activated. Once the sensor warms up, it can detect heat and movement in the surrounding areas, creating a protective "grid." If a moving object blocks too many grid zones and the infrared energy levels change rapidly, the sensors are tripped. Passive infrared sensors are sensitive to a person's skin temperature through emitted black body radiation at mid infrared wavelengths, in contrast to background objects at room temperature. No energy is emitted from the sensor, thus the name "passive infrared" (PIR). This distinguishes it from the electric eye for instance (not usually considered a "motion detector"), in which the crossing of a person or vehicle interrupts a visible or infrared beam.

The people counting system based on PIR motion detectors can be presented. For each passage monitored, three PIR sensors are installed at a distance of 0.8m.. Sensors detect motion events and transmit these data to the microcontroller which infers a people count from correlating the number, phase and time difference of peaks found in the signal. The system achieves a rate of 100% to detect the direction of movement, and accurately detects 89% of the number of people passing. PIR sensors provide an alternative to IR sensor arrays, however the cost and effort of employing multiple sensor nodes for each entry/exit point is a cost-side disadvantage. The goal of this thesis is to develop a system based on just one PIR sensor and one sensor node per each observed entry/exit point.

The human body radiates infrared waves with wavelengths of 8 to 12 micrometers. Any movement by a person leads to a change in the amount of infrared energy which a sensor can detect within its range. The PIR sensor reacts to this change in infrared energy and provides a

Low frequency, small amplitude signal.

Advantages

1. A passive infrared sensor can be set up to ignore animals up to a certain weight hence it is immune to pets. They can also be set up to ignore a large animal or multiple small animals.
2. They are small, inexpensive, low-power, easy to use and don't wear out.

2.5 Sensor Fusion

It is a form of a building occupancy estimation system by applying different types of sensors i.e. camera, CO_2 and PIR sensors. It uses a method called **Hidden Markovian Model (HMM)** based on an Extended Kalman Filter (EKF) in order to derive a building occupancy. The approach integrates historical data and current sensor readings to estimate the true state of the system, adjusting for sensor noise (false observations) and stochastic processes, e.g. uncertain people movement patterns.

2.6 Microwave (MW):

Its system sends out microwave pulses and measures the reflection off a moving object. These detect motion through the principle of Doppler radar where a continuous wave of microwave radiation is emitted, and phase shifts in the reflected microwaves due to motion of an object toward (or away from) the receiver result in a heterodyne signal at low audio frequencies.

Advantages 1. They cover a larger area than infrared sensors

Disadvantage 1. They are vulnerable to electrical interference and are more expensive

2.7 Dual Technology Motion Sensors:

This technology involves the motion sensors having combined features in an attempt to reduce false triggering. For example, a passive infrared (PIR) sensor could be combined with a microwave sensor hence since each operates in different areas of the spectrum, and one is passive and one is active. The Dual Technology motion sensors are not as likely as other types of sensor to cause false detection (e.i to be triggered) since both of the sensors have to be tripped. However, this does not mean that they will never have false detection. Many modern motion detectors use combinations of different technologies.

A dual element Passive-Infrared (PIR) sensor is used with a rotating slit aperture to map a narrow scanning beam on the sensing elements through each lens of a Fresnel lens array. The stimuli generated due to each thermal object fall in the active Fresnel zones in a certain direction based on their locations and temperature variations on the surfaces of the sources. These signals are used to analyze the signatures of stationary thermal objects, their slight movements and thermal field gradient changes of the source surfaces provided the object projected area is less than the area of the active zone. Pattern matching is performed using Dynamic Time Warping (DTW) algorithm on STFT reduced length time vectors. The object space is divided into m-active zones that correspond to the Fresnel zones in a Fresnel lens array. Within passive IR region from each of the active zones, the system identifies not only the heat intensity changes but also detects the slight movement of the thermal source. The efficiency of the system is dependent on the number of active Fresnel zones and the angular separation between them.

Advantage

1. While combining multiple sensing technologies into one detector can help reduce false triggering, it does so at the expense of reduced detection probabilities and increased vulnerability. For example, many dual-tech sensors combine both a PIR sensor and a microwave sensor into one unit. In order for motion to be detected, both sensors must trip together. This lowers the probability of a false alarm since heat and light changes may trip the PIR but not the microwave, or trees may trigger the microwave but not the PIR. If an intruder is able to fool the PIR or microwave, however, the sensor will not detect. Dual-tech sensors are only as strong as their weakest link.

2. The PIR technology will be paired with another model to maximize accuracy and reduce energy usage. PIR draws less energy than microwave detection and so many sensors are calibrated so that when the PIR sensor is tripped, it activates a microwave sensor. If the latter also picks up an intruder, then the alarm is sounded

2.8 Ultrasonic:

It sends out pulses of ultrasonic waves and measures the reflection off a moving object. The ultrasonic transducers are the sensors that are used in ultrasonic motion detector as it can even detect motion in areas where there are not supposed to be any moving object. The ultrasonic motion detectors have two transducers: one that emits an ultrasonic wave and the other that picks up reflections from the different objects in the area. The reflected waves arrive at the receiver in constant phase if none of the objects in the area are moving. If something moves then the received signal will be shifted in phase hence the phase comparable detects the shifted phase and sends a triggering pulse to the output

A system employing ultrasonic sensors per each observed area with three node sensor cluster can be established whereby each sensor node mounts an ultrasonic sensor. Nodes in each cluster communicate the sensor readings to the cluster's coordinator node. The latter contributes its own sensor measurements. By means of a distributed algorithm, nodes decide on whether to count a detected person. The sensor nodes require clock synchronization at the millisecond level in order to correlate the data exchanged.

Advantages

1. It is very sensitive and extremely fast in action
2. It can also be used to detect people since they are inexpensive and can operate with a narrow beam widths.

Disadvantage

1. It sometimes responds to normal environment vibration that can be caused by a passing car or a plane overhead.
2. Its installation options are also limited since the ultrasonic beams are easily blocked by thin material like a paper therefore reflections from blowing curtains, pets and flying insects can easily cause false detection.
3. The sensor can be sensitive to motion in areas where coverage isn't desired, for instance, due to reflections of sound waves around corners

2.9 Vibration:

Detects vibration. These can be purchased or easily made at home. A homemade vibration sensor uses a small mass on a lever, which is activated by a switch to an alarm when it vibrates. Homemade motion sensors can work, but they can also be unreliable.

2.10 Video Motion Sensors:

It combines video cameras with advanced signal processing. A simple algorithm for motion detection by a fixed camera compares the current image with a reference image and simply counts the number of different pixels. Since images will naturally differ due to factors such as varying lighting, camera flicker, and CCD dark currents, pre-processing is useful to reduce the number of false positive detection's. More complex algorithms are necessary to detect motion when the camera itself is moving, or when the motion of a specific object must be detected in a field containing other movement which can be ignored.

When counting people using multiple video cameras the focus lies on extracting the size and moving patterns of individuals passing. By means of motion histograms based on frame-differenced images, the histograms classify detected movements. Probabilistic correlation is applied to determine a people count. The results of multiple cameras are joined in order to form a movement vector for each individual recognized while with a single ceiling-mounted camera, which identifies people by background extraction of the camera image. A non-background “blob” is recognized, and its size is estimated and compared to previously established bounds of people’s pixel dimensions. A people count is derived from the results of this analysis. The system reaches a claimed accuracy of 98.5%.

Disadvantage

1. It requires an ambient light source and relatively powerful compute resources to perform image processing.
2. It is very expensive to buy the cameras.
3. Intefacing the camera is not simple

2.11 Tomographic motion detector (radar)

Tomographic motion detection systems sense disturbances to radio waves as they pass from node to node of a mesh network. They have the ability to detect over complete areas because they can sense through walls and obstructions. Conventional radio detection and ranging (radar) uses beamed and reflected microwave energy to detect, locate, and track objects over distances of many miles

The method exploits the fact that a human body interferes with radio signals by introducing irregularities in the radio signature which indicate possible human presence. By analyzing radio signals between radios transceivers embedded in 2.4 GHz wireless power outlets, the original environment is not visually modified and a certain level of sensorial intelligence is introduced without additional sensors. The analysis of the signal strength variations in principal components space enhances the detection accuracy level of human presence detection method.

Disadvantage.

1. Commercial use has been limited primarily because most radar systems are large, and they can be complex and cost $40,000 or more.

2.12 Photoelectric beams and lasers

They requires an emitter(to generate light) and a receiver that sees the light from the emitter. Most of them uses LED as a light source eg a solid state semiconductor (similar to a diode) except that it emits a small amount of light when current flows through it in the forward direction.

advantages

- ✓ Some have built in countdown timer which turns off the lights after a predefined time period hence saves on energy. The lighting is needed only when someone is present
- ✓ Some uses a built-in light sensor where light will not be ON when the monitored area is bright enough even if movement is detected there.

Disadvantages

- ✧ The beams are problematic in areas where objects are moved frequently since they cannot see through obstructions that blocks the beams.
- ✧ The beams must be carefully installed (alignment) and maintained on regular basis as the beams emitter or detector gets dumped
- ✧ It may require recalibration and manual adjustment to get the emitter pointing back to the detector after sometimes.

2.13 METHOD SETTLED UPON

Overally the pyroelectric infrared (PIR) motion sensors type is the best practical means of detecting the human body without contacts based on the detection performance, noise resistance, causes of false detection and performance cost. The have been extensively used in indoor and outdoor applications since they are low cost, easy to use and widely available. For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. Note that PIRs won't tell you how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance (although it can be hacked somewhere) and they are also sometimes set on by pets.

Since the passive infrared motion detectors can be falsely triggered by warm air movement or other disturbances that can alter the infrared radiation levels in the area. The problem is prevented in newer systems by using two or more infrared sensors that monitor different zones within a protected area.

The sensor circuitry is such that apart from directly using the output of a classical PIR sensor, an analog signal is extracted from the PIR output and then sampled. The sampling leads into intelligent signal processing algorithms development by using the discrete time sensor signal. This method makes it possible to develop human, pet and flame or just hot body detection methods. It is also possible to find the direction of moving objects i.e. moving to or from the entrance by estimating their distance from the sensor at any specific time. This path of a moving target can be estimated using a PIR sensor array.

2.13.1 Points to consider when selecting a sensor

- ✓ The presence of a person must be detected over a wide area.
- ✓ The detection area must be limited so that only the person at the entrance or exit (door) is detected.
- ✓ The detection range should extends to the walls
- ✓ Non moving people or objects without a temperature difference are not to be detected
- ✓ Detection performance, noise resistance, causes of false detection and performance cost

2.13.2 Sensor specifications

- ✓ **Resolution :** resolution is the smallest measurement a sensor can reliably indicate. higher resolutions in sensor design is primarily a fight against electrical noise. Bandwidth (frequency response) indicates how sensors respond at different frequencies.A low-pass filter will reduce or eliminate high frequency noise, while reducing the sensor's bandwidth. Low-pass filtered signals have less noise and therefore better resolution but at the expense of usable bandwidth.
- ✓ **Accuracy.** It is sensitive to the measured quality only.
- ✓ **Range.** It should have a wide range of values it can read.
- ✓ **Reliability**
- ✓ **Performance.** It does not influence the measured property.
- ✓ **Supportability**
- ✓ **Interfaces.** It should be easy to interface.
- ✓ **Design constraints**

2.13.3 Advantages of PIR Sensors

- ✓ Relatively Small in size
- ✓ Inexpensive compared to other motion sensors like video
- ✓ Easy to use since they are easily interfaced
- ✓ They don't wear out i.e they are rugged
- ✓ Digital signal processing (DSP)
- ✓ Power adjustable, save more energy
- ✓ Two-way differential high impedance sensor input
- ✓ Built-in filter, screen the interference by other frequency
- ✓ Excellent power supply rejection, Insensitive to RF interference

- ✓ Schmidt REL output
- ✓ Low voltage, low power consumption(i.e between 2.7V to 3.3V), instantaneous settling after power up

2.14 PYROELECTRONIC INFRARED (PIR) SENSOR

2.14.1 Infrared Radiation

They exists in the electromagnetic spectrum at a wavelength that is longer than visible light so it cannot be seen but it can be detected. Objects that generate heat also generate infrared radiation and those objects include animals and the human body whose radiation is strongest at a wavelength of **9.4um.** An unprocessed silicon wafer makes a good IR window in a weatherproof enclosure for outdoor use. It also provides additional filtering for light in the visible range. 9.4um infrared will also pass through polyethylene which is usually used to make Fresnel lenses to focus the infrared onto the sensor elements.

2.14.2 Pyroelectric Sensors

They are often referred to as PIR, "Passive Infrared", or "IR motion" sensors. The PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. PIRs are basically made of a pyroelectric sensor which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in motion detection is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

The pyroelectric sensor is made of a crystalline material that generates a surface electric charge when exposed to heat in the form of infrared radiation. When the amount of radiation striking the crystal changes, the amount of charge also changes and can then be measured with a sensitive FET device built into the sensor. The sensor elements are sensitive to radiation over a wide range so a filter window is added to the package to limit detectable radiation to the 8 to 14mm range which is most sensitive to human body radiation.

Figure1. Spectral Response of PIR SENSOR Window Materials

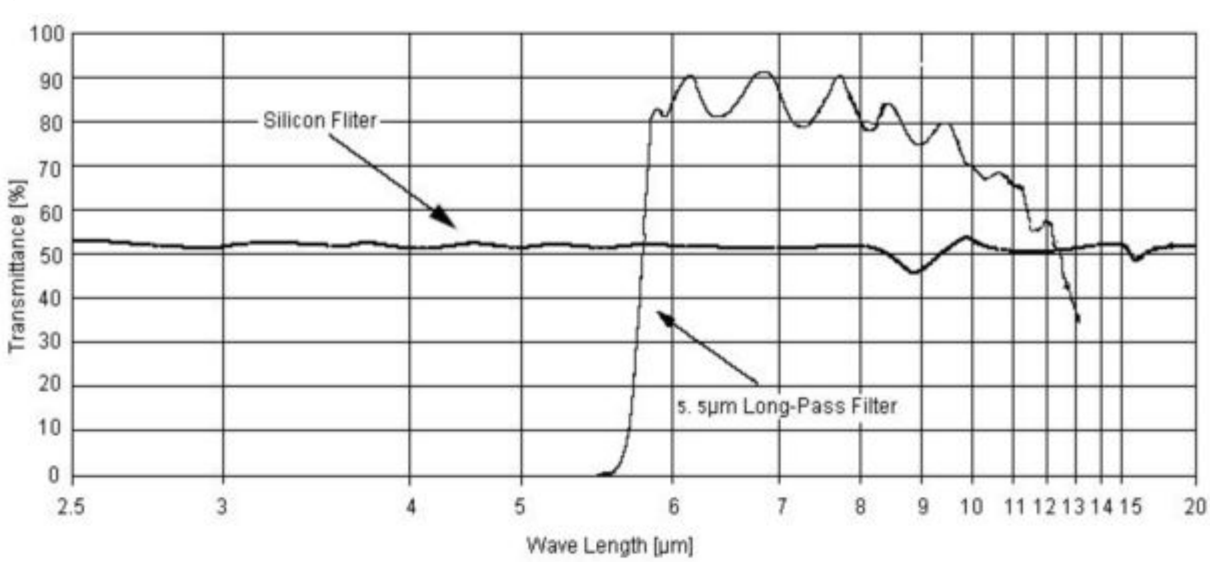
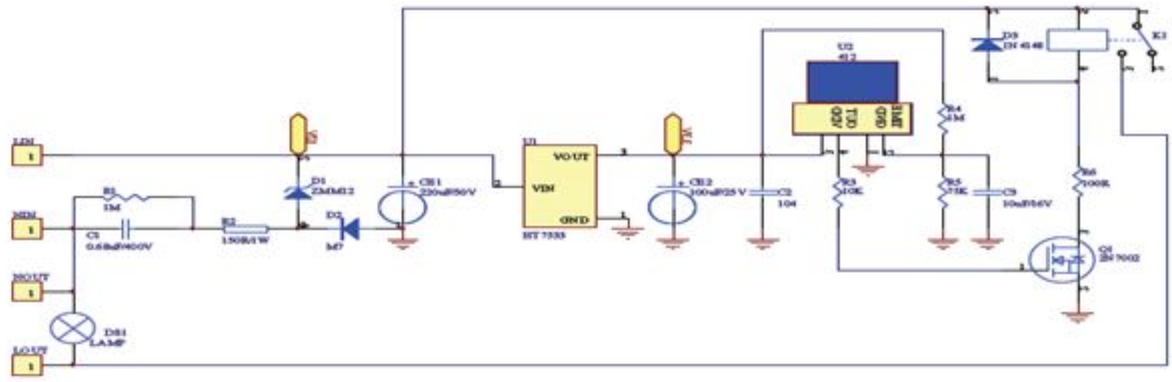


FIGURE 2. The reference circuit of the Am412 PIR Sensor.



Typically, the FET source terminal pin 2 connects through a pull-down resistor to ground and feeds into a two stage amplifier having signal conditioning circuits. The amplifier is typically bandwidth limited to below 10Hz to reject high frequency noise and is followed by a window comparator that responds to both the positive and negative transitions of the sensor output signal. A well filtered power source of from 3 to 15 volts should be connected to the FET drain terminal pin 1.The PIR325 sensor has two sensing elements connected in a voltage bucking configuration. This arrangement cancels signals caused by vibration, temperature changes and sunlight. A body passing in front of the sensor will activate first one and then the other element whereas other sources will affect both elements simultaneously and be cancelled. The radiation source must pass across the sensor in a horizontal direction when sensor pins 1 and 2 are on a horizontal plane so that the elements are sequentially exposed to the IR source

2.14.3 Theory of operation of pyroelectronic infrared (pir) sensor

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

The IR sensor itself is housed in a hermetically sealed metal can to improve noise /temperature /humidity immunity. There is a window made of IR-transmissive material (typically coated silicon since that is very easy to come by) that protects the sensing element. Behind the window are the two balanced sensors.

2.14.3.1 Lenses

PIR sensors are rather generic and for the most part vary only in price and sensitivity. Most of the real magic happens with the optics. This is a pretty good idea for manufacturing: the PIR sensor and circuitry is fixed and costs a few dollars. The lens costs only a few cents and can change the breadth, range, sensing pattern, very easily.

In the diagram up top, the lens is just a piece of plastic, but that means that the detection area is just two rectangles. Usually we'd like to have a detection area that is much larger. To do that, we use a simple lens such as those found in a camera: they condenses a large area (such as a landscape) into a small one (on film or a CCD sensor). For reasons that will be apparent soon, we would like to make the PIR lenses small and thin and moldable from cheap plastic, even though it may add distortion. For this reason the sensors are actually Fresnel lenses

2.15.2 THE GENERAL BLOCK DIAGRAM OF PIR SENSOR

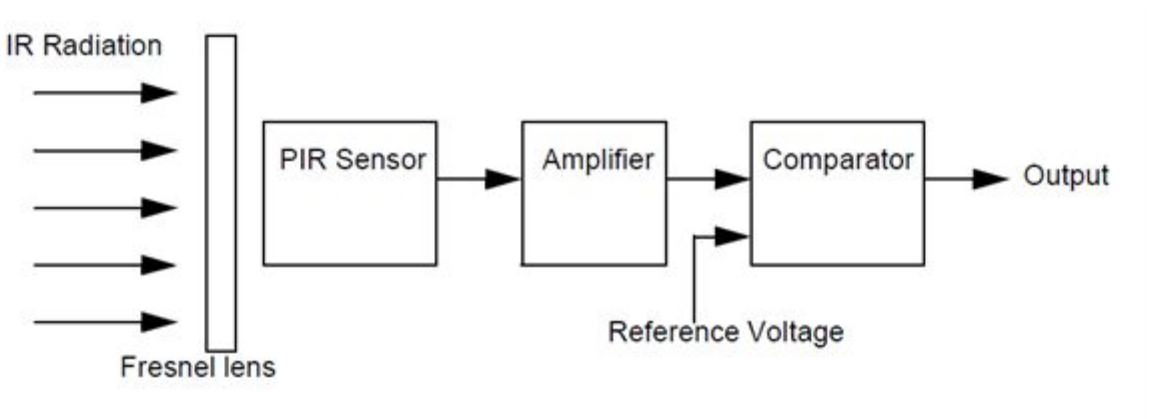
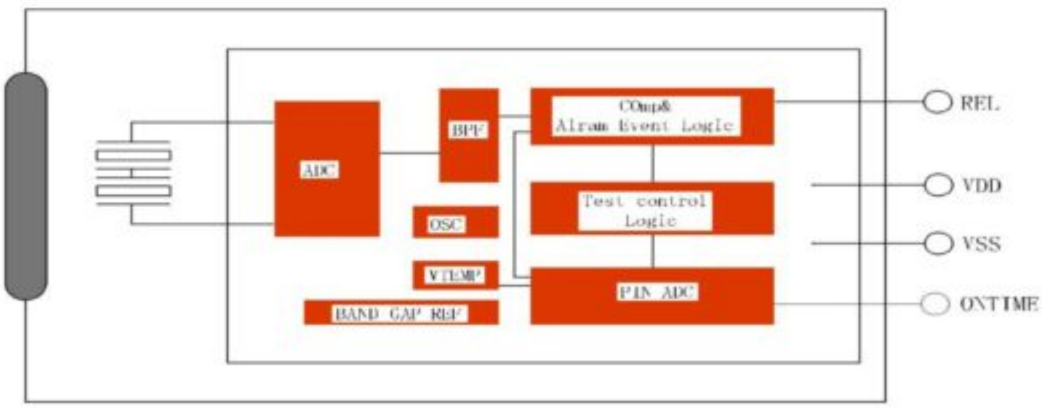


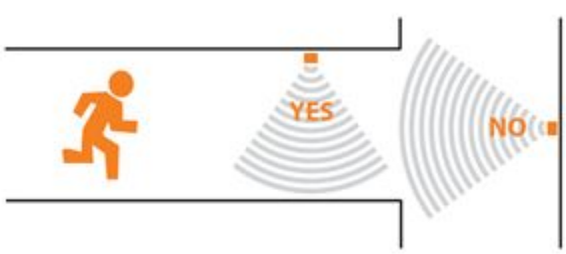
FIG2. THE INTERNAL BLOCK DIAGRAM OF PIR 412 SENSOR



NB Everything emits some low level radiation and the hotter something is, the more radiation it emits.

2.15 Best Practices for Mounting Sensors

Keep PIR sensors 10—15 feet away from heating vents, where the sunlight shines in, and radiators. If a motion sensor detects a swift change in heat, even that of a cloud passing quickly over direct sunlight shining into your living room, it could be tripped. Motion sensors work best when the detected body walks in parallel to the sensor, not toward it. For example, in a hallway you tend to walk parallel to the walls, not directly toward them. Find walls where the movements would be alongside like a hallway or narrow pathway that leads to a room.

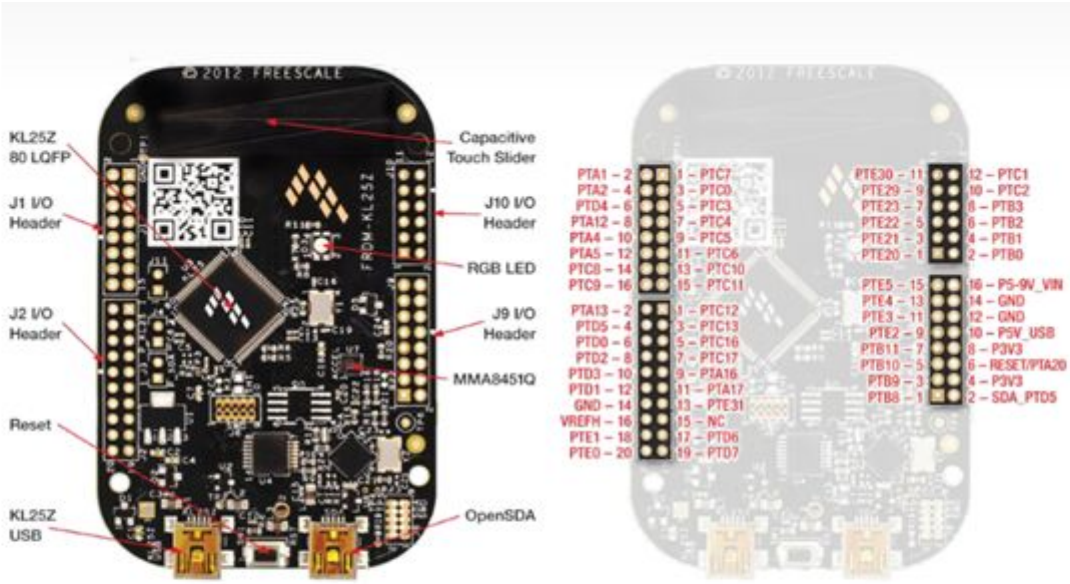


2.16 Other Applications of Motion Sensors

- To open and close automatic Doors
- To turn on and off automatic water faucets and toilets
- To turn on lights when a person enters a room
- To control ATM displays
- At automatic ticket gates
- For some parking meters

2.17 MICROCONTROLLER

THE PIN ARRANGEMENT OF THE USED MICROCONTROLLER



2.17.1 POWER CHARACTERISTICS

- **Supply Voltage (V_{cc}):** The value as specified by level (Min-Typ-Max) of the direct supply voltage, applied to an IC.
- **Supply Current :** The continuous current (in A) required by this item during normal operation.
- **Power Dissipation :** The maximum permissible power dissipation per output (in W) of this item at specified ambient temperature

2.17.2 TIMERS

The timer type(s), and the number of bits and number of channels provided.

NUMBER OF BITS	THE NUMBER OF BITS OF THE TIMER
8 Bits	The microcontroller has a 8-bit timer
16 Bits	The microcontroller has a 8-bit timer
24 Bits	The microcontroller has a 8-bit timer
32 Bits	The microcontroller has a 8-bit timer

It also specify the number of timers embedded in the microcontroller

2.17.3 INTERFACES AND PORTS

Connection to a microcontroller that provides a data path between the microcontroller and external devices, such as a keyboard, display, or reader. It may provide input only, output only, or both input and output .**Serial Interfaces** The serial interface type(s), and the number of channels provided.**The Controller Area Network (CAN)** protocol is a serial communication protocol originally developed by Robert Bosch for use in serial communication networks in vehicles. Several major auto manufacturers are either using CAN networks in their vehicles or are developing them for future vehicles.**Inter-Integrated Circuit (I²C)** is a bus is an inexpensive type of chip interconnection that is popular on circuit boards .**The Serial Peripheral Interface (SPI)** is similar to the SCI, although it is used to communicate synchronously over shorter distances at up to 4 Mbit/s. The SPI allows the microcontroller to communicate with peripheral devices, which could be anything from a simple TTL shift register to a complete subsystem such as an LCD display or an A/D converter system. The SPI is flexible enough to interface directly with numerous standard peripherals from many manufacturers. SPIs can also be used to expand the number of inputs and outputs of the microcontroller with the minimum number of pins. **The Serial Communications Interface (SCI)** is an independent serial I/O subsystem (full-duplex UART-type asynchronous system). The SCI can be used for communications between the microcontroller and a terminal, PC, or other microcontrollers in the form of a network. An on-chip baud rate generator derives standard baud-rate frequencies from the microcontroller oscillator.**UART g**enerically, this refers to any device capable of implementing a variety of asynchronous serial protocols, such as RS 232, HDLC and SDLC. In this context, it refers to the operating mode of the SCCs that provides this functionality.**USART** is a full duplex synchronous / asynchronous receiver and transmitter that supports a wide range of software programmable baud rates and data formats. Then finally (**USB**)Universal Serial Port.

2.17.4 SERIAL PORT CHANNELS:

The number of serial ports the microcontroller has. The MKL25Z128VLK4 in an 80 LQFP package

NUMBER OF I/O PORTS : The total number of input, output, and I/O ports combined. An I/O port is a connection to a CPU that is configured or programmed to provide data path between the CPU and external devices such as a keyboard, display, or reader; it may be an input port or an output port, or it may be bi-directional.

2.17.5 CONVERTERS

The resolution of the A/D and D/A converters, and the number of channels provided.

It can either have 8-bits timer, 16-bits timer or 32-bits timer

Number of A/D Converters : A/D converters take analog inputs and convert them to digital signals. For example, a sine wave input can be converted into a series of numbers that represent the value of the input curve at any given point on the curve. The resolution of the converter is determined by the number of bits (the more the better) and the sample rate (the faster the better).

2.17.6 OTHERS

1. Operating Range : it specifies the range of operation where the microcontroller can be used i.e industrial, commercial, military or automotive

2. Pin Count : The number of pins the DSP has. The question is presented as a range for convenience of selection.

3. Operating Temperature: The value as specified by level of the ambient temperature (in degree Celsius) in which this microcontroller was designed to operate.

4. Watchdog Timer: A watchdog timer is a simple countdown timer, which is used to reset a microprocessor after a specific interval of time.

5. DMA Channels : *Direct Memory Access (DMA is a* technique for transferring data directly between two peripherals (usually memory and an I/O device) with only minimal intervention by the processor.

6. PWM Channels : Pulse Width Modulation (PWM) is a technique for controlling analog circuits with a processor's digital outputs. PWM is employed in a wide range of applications, from measurement and communications to power control and conversion.

7. Power Saving Modes : Functions that allow the microcontroller to same power. Some of these functions include: Wait Mode, STOP Mode, SLEEP Mode, etc.

8. Oscillator Safeguard (OSG) : The Oscillator Safeguard (OSG) feature provides the following functions to the MCU: Filters spikes in the oscillator lines, Manages the Low Frequency Auxiliary Oscillator (LFAO), and Automatically limits the internal frequency (f_{INT}) clock frequency as a function of supply voltage.

9. Low Frequency Auxiliary Oscillator (LFAO) : A backup oscillator.

10. The Low Voltage Detector (LVD) allows the MCU to be used without any external RESET circuitry. If the LVD is not used, an external circuit is mandatory to ensure correct Power On Reset operation.

11. The Readout Protection feature is used to protect the program memory against external readout. Once the Readout Protection is active, it is no longer possible to gain access to the OTP or ROM content.

12. LCD Driver : A feature used to connect and control external LCDs.

13. Data Bus : The number of bits the data bus can transfer at the same time. A bidirectional set of conductive paths on which data or instruction codes are transferred into the DSP, or on which the result of an operation or computation is sent out from the DSP.

14. Life Cycle Stage: This is the current product life cycle stage

15. Clock Speed: The speed of the microcontroller clock. The maximum frequency of the device.

16. Number of Interrupts : An asynchronous electrical signal from a peripheral to the processor. When the peripheral asserts this signal, we say that an interrupt occurs. When an interrupt occurs, the current context is saved and an interrupt service routine is executed. When the interrupt service routine exits, control of the processor is returned to whatever part of the software was previously running.

2.17.7 MEMORY

- **Internal RAM Size :** Random access memory (RAM) can be read from or written to in a nonlinear manner. RAM derives its name from the fact that any byte of memory can be accessed randomly instead of sequentially. RAM does not retain data in memory when power is removed.

- **Internal ROM Type :** The type of the on-chip instruction ROM. Eg **ROMLess are** Devices do not include read-only memory (ROM).**Mask ROM** is static, read-only memory (ROM) that is programmed during fabrication with a special mask that contains the customer code. The code cannot be modified afterwards. Manufacturers that produce large semiconductors often use Mask ROM because it is the most cost-effective ROM available.**Erasable programmable read-only memory (EPROM)** is a type of PROM that can be erased through exposure to ultraviolet light and then reprogrammed.**Electrically erasable programmable read-only memory (EEPROM)** is a type of PROM that can be erased electrically and then reprogrammed.Flash is a form of electrically erasable, programmable, read-only memory (EEPROM) that can be erased and reprogrammed in blocks instead of one byte at a time. **Flash memory** is non-volatile, which means that it does not need a constant power supply in order to retain data. Flash offers extremely fast access times, low power consumption, and relative immunity to severe shock or vibration. Flash memory chips have a lifespan of approximately 100,000 write cycles - a fact that makes Flash unsuitable for use as computer main memory. Typically, Flash memory chips are used in portable or compact devices such as digital cameras, cell phones, pagers, and scanners. Flash memory chips are also used as solid-state disks in laptops and as memory cards for video game consoles.**Factory advanced service technique read-only memory (FASTROM)** is pre-programmed by the manufacturer with the customer's code and selected options. Compared to Flash, FASTROM provides improved programming efficiency for large quantities. Compared to Mask ROM, FASTROM offers shorter lead-times and reprogrammability. And One time programmed memory **(OTP)**.

CHAPTER 3 METHODOLOGY

3.1 INTRODUCTION

It involves the following processes .

1.Define the problem –it involves clarifying the objectives, defining the concerned issues, and limiting the problem so that it can be effectively studied.

2. Identify feasible alternatives – All the alternatives should be considered to make sure that the best approach is chosen.

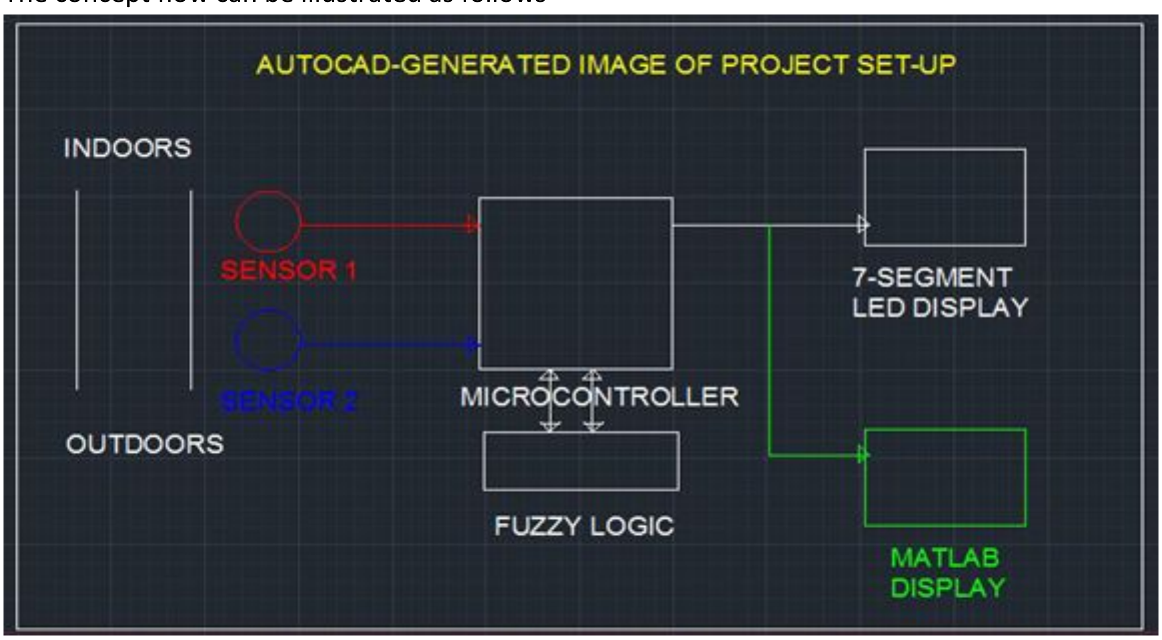
3. Select the evaluation criteria – The criteria for the evaluation process can vary considerably, so the appropriate ones must be chosen.

4. Applying modeling techniques – A model or series of models should be used.

5. Generate input data – The requirements for appropriate input data should be specified.

6. Manipulate the model – After data is collected and inputed, the model may be used. Analysis after using the model will lead to recommendation for some kind of action.

The concept flow can be illustrated as follows



The principal of operation

The two sensors will collectively collect data based on the level of infrared amount detection at the door, these results are temporarily stored on the microcontroller for about 2 seconds. Now based on the pattern received and the expected pattern, a decision is made by help of fuzzy rules.

No.	Real Time	Time (ms)	Sensor1 reading	Sensor2 reading	Movement
1	**6:00:00:001**	**1**	**0**	**0**	
2	**6:00:00:020**	**20**	**1**	**0**	
3	**6:00:00:200**	**200**	**1**	**1**	
4	**6:00:00:800**	**800**	**0**	**1**	
5	**6:00:01:200**	**1200**	**0**	**0**	**Entry**

6	**6:00:01:400**	**1400**	**0**	**1**	
7	**6:00:02:100**	**2100**	**1**	**1**	
8	**6:00:02:400**	**2400**	**1**	**0**	
9	**6:00:02:600**	**2600**	**0**	**0**	**Exit**
10	**6:00:03:000**	**3000**	**1**	**1**	
11	**6:00:03:600**	**3600**	**0**	**0**	**Blockage**

From the above table looking at the readings received from the two sensors, a binary pattern can be realized from No. 2 at 20ms to No. 5 at 1200ms (red in color). From my sensors arrangement the pattern is recognized as **entry** into the room. From the real time column, somebody entered into the building at 6:00:01:200 am . The blue color pattern from No.6 to No.9 shows an **exit** from the room at 6:00:02:600 am. The green color pattern at No.10 shows some **blockage** at the sensors field of view hence an alarm can be triggered at this intense. It's time is also recorded as 6:00:03:600 am .

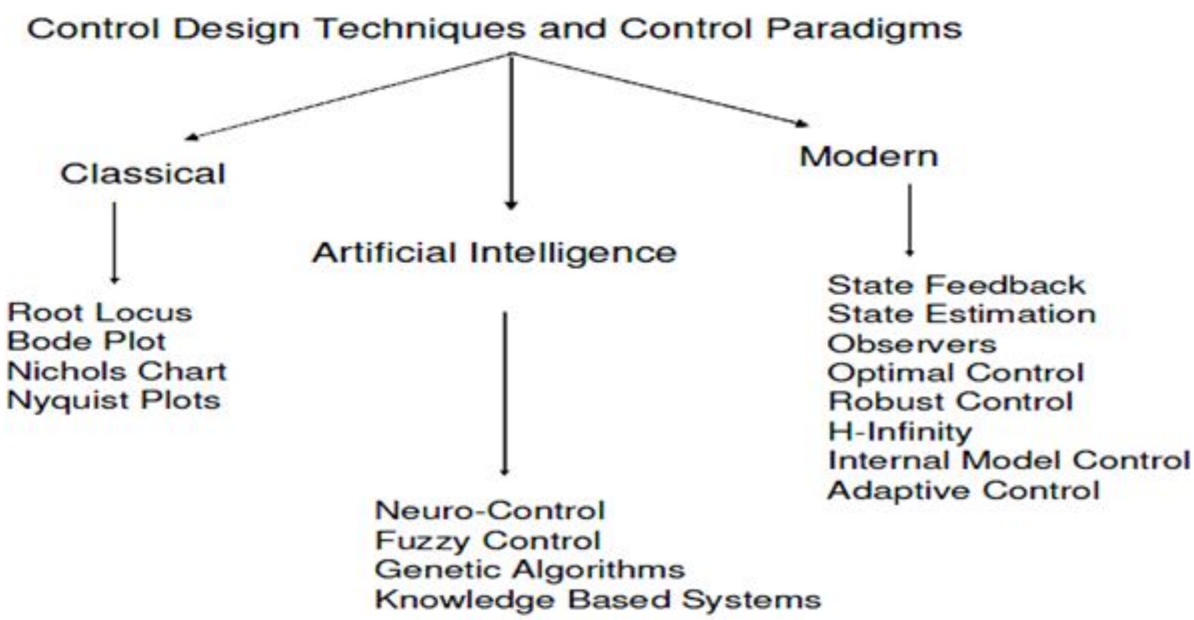
The fuzzy **if... Then....** Rules

1. If the sensors pattern is 011110 then the movement is entry
2. If the sensors pattern is 101101 then the movement is exit
3. If the sensors pattern is 11 then the movement is blockage
4. If the sensors pattern is 00 then there is no movement

Every pattern $\boldsymbol{00011110_2 = 1E_{16}}$ leads to increase in the number of people in the room as the counter will increment while for every pattern $\boldsymbol{00101101_2=2D_{16}}$ leads to a decrease in the number of people in the room hence the counter will be decremented.

The above logic will work perfectly well for a single entry/exit door i.e one movement at a time.

3.2 Where is fuzzy in control design?



3.3 Fuzzy Control System Design

3.3.1 Introduction

Nowadays fuzzy logic control has emerged as one of the most active and fruitful research areas in the application of fuzzy set theory, fuzzy logic and fuzzy reasoning.Many industrial and consumer products using fuzzy logic technology have been built and successfully sold worldwide. In contrast to conventional control techniques, fuzzy logic control is best utilized in complex ill-defined processes that can be controlled by a skilled human operator without much knowledge of their underlying dynamics.

The basic idea behind fuzzy logic control is to incorporate the "expert experience" of a human operator in the design of a controller in controlling a process whose input-output relationship is described by a collection of fuzzy control rules (e.g. IF-THEN rules) involving linguistic variables. This utilization of linguistic variables, fuzzy control rules, and approximate reasoning provides a means to incorporate human expert experience in designing the controller.

Fuzzy Logic provides a completely different, unorthodox way to approach a control problem. This method focuses on what the system should do rather than trying to understand how it works. One can concentrate on solving the problem rather than trying to model the system mathematically, if that is even possible. This almost invariably leads to quicker, cheaper solutions. Once understood, this technology is not difficult to apply and the results are usually quite surprising and pleasing.

3.3.2 Benefits of Fuzzy Control

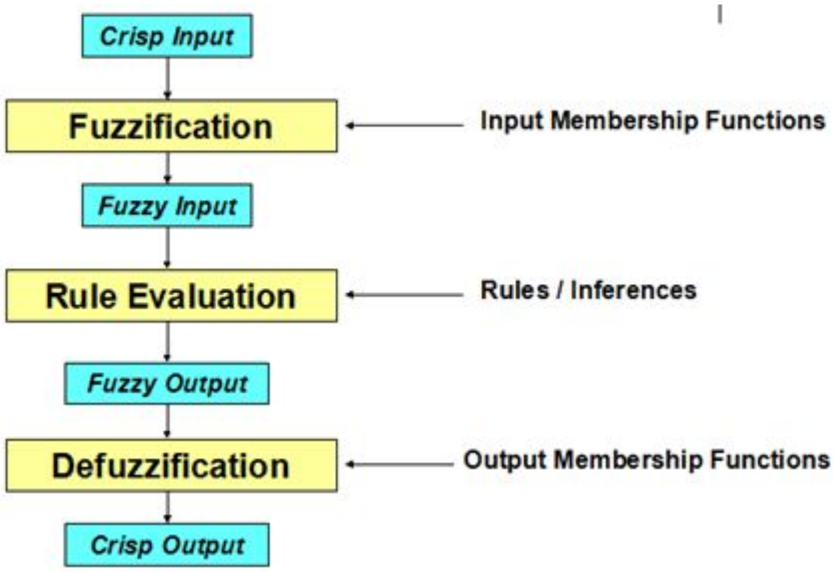
- Suitable for Complex Ill-defined Systems where mathematical modeling is difficult or Plant is too abstract e.g. Complex chemical plant, cementkiln, etc.
- Able to design along linguistic lines – usage of rules based on experience
- Better performance than Conventional PID Controllers
- Simple to design

3.3.3 BASIC STRUCTURE OF A FUZZY LOGIC CONTROLLER

The basic structure of a fuzzy logic controller (FLC) comprises the following four primary components

1. fuzzification interface
2. knowledge base,
3. decision logic, and
4. defuzzification interface.

FIGURE 3.1 THE OPERATIONS OF THE FUZZY LOGIC



3.3.3.1 Preprocessing (Crisp Input)

The inputs are crisp measurement from the sensor measuring equipment rather than linguistic. A preprocessor is the first block which shows the conditions the measurements before enter the controller.

A.1 The fuzzification interface performs the following functions:

(a) Measures the value of the input variable,
(b) conducts scale mapping,
(c) converts data into linguistic values that can be denoted using fuzzy sets according to the fuzzification

Therefore the stages function is to converts each piece of input data to degrees of membership by a looking up in one or several membership functions.

3.3.3.2. The knowledge base includes a database and a linguistic control rule base.

(a) The database provides the required definitions to define linguistic control rules and to facilitate the FLC's processing of fuzzy data
(b) the rule base describes the control target and control strategies of domain experts according to linguistic control rules.

The collection of rules is called a rule base. The rules are in "*If Then*" format and formally the *If side* is called the *conditions* and the *Then side* is called the *conclusion.* The computer is able to execute the rules and compute a control signal depending on the measured inputs *error* (e) and *change in error.*(dE). In a rule based controller the control strategy is stored in a more or less natural language. A rule base controller is easy to understand and easy to maintain for a non- specialist end user and an equivalent controller could be implemented using conventional techniques.[6]

The fuzzy **if... Then....** Rules

5. If the sensors pattern is 011110 then the movement is entry
6. If the sensors pattern is 101101 then the movement is exit
7. If the sensors pattern is 11 then the movement is blockage
8. If the sensors pattern is 00 then there is no movement

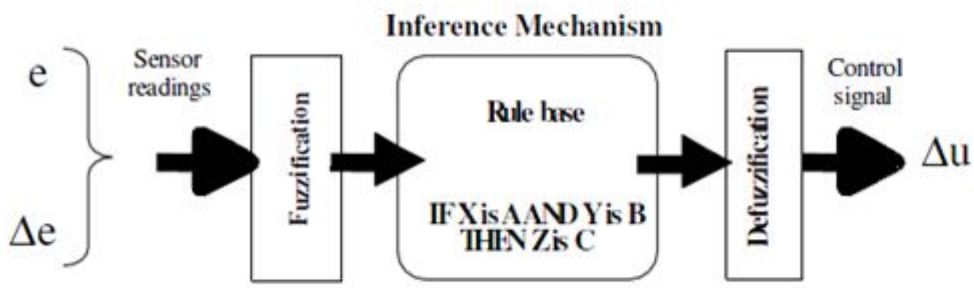
3.3.3.3 The defuzzification interface performs the following functions:

(a) Scale mapping

(b) defuzzification by computing solutions based on the inferred fuzzy-control actions, thereby obtaining crisp control actions.

Defuzzification is when all the actions that have been activated are combined and converted into a single non-fuzzy output signal which is the control signal of the system. The output levels are depending on the rules that the systems have and the positions depending on the non-linearities existing to the systems. To achieve the result, develop the control curve of the system representing the I/O relation of the systems and based on the information; define the output degree of the membership function with the aim to minimize the effect of the non-linearity.

The event flow on fuzzy diagram



Preprocessing is done since the inputs are most often hard or crisp measurements from a measuring equipment(sensor readings) rather than linguistic,it conditions the measurements before they enter the controller. A quantizer is necessary to convert the incoming values in order to find the best level in a discrete universe.

3.4 Implementation of fuzzy logic

3.4.1Fuzzy Logic Toolbox

There are five primary graphical user interface (GUI) tools for building, editing and observing fuzzy inference systems in the toolbox:-

1. Fuzzy Inference System (FIS) editor
2. Membership Function editor
3. Rule Editor
4. Rule Viewer
5. Surface Viewer

These GUI are dynamically linked and if the changes make to the FIS to the one of the toolbox, the effect can be seen in other GUIs. In addition to these five primary GUIs, the toolbox includes the graphical ANFIS Editor GUI, which is used for building and analyzing Sugeno-types adaptive neural fuzzy inference systems.

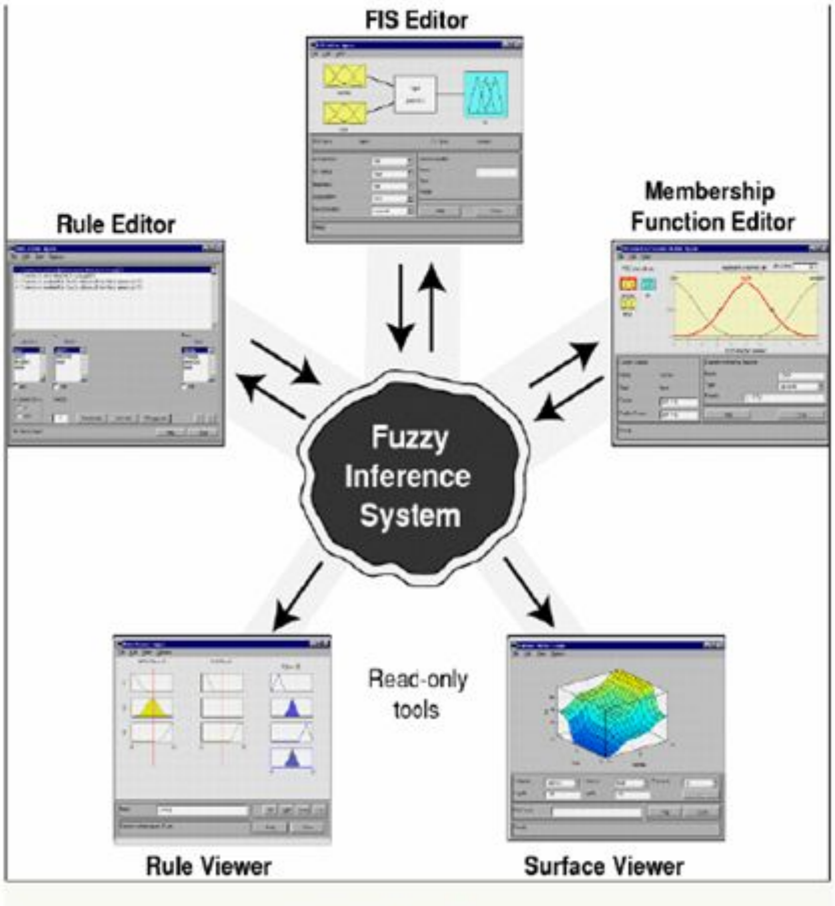


Figure 4: Fuzzy Inference System

3.4.1Fuzzy Inference System

Fuzzy inference systems (FIS) are also known as fuzzy rule-based systems, fuzzy model, fuzzy expert system, and fuzzy associative memory. The FIS formulates suitable rules and based upon the rules the decision is made. This is mainly based on the concepts of the fuzzy set theory, fuzzy IF–THEN rules, and fuzzy reasoning. FIS uses "IF. . . THEN. . . " statements, and the connectors present in the rule statement are "OR" or "AND" to make the necessary decision rules. The basic FIS can take either fuzzy inputs or crisp inputs, but the outputs it produces are almost always fuzzy sets. When the FIS is used as a controller, it is necessary to have a crisp output. Therefore in this case defuzzification method is adopted to best extract a crisp value that best represents a fuzzy set.

3.4.1.1 Construction and Working of Inference System

Fuzzy inference system consists of a fuzzification interface, a rule base, a database, a decision-making unit, and finally a defuzzification interface.

The FIS works as follows i.e the crisp input is converted in to fuzzy by using fuzzification method. After fuzzification the rule base is formed. The rule base and the database are jointly referred to as the *knowledge base*. Defuzzification is used to convert fuzzy value to the real world value which is the output.

The steps of *fuzzy reasoning* (inference operations upon fuzzy IF–THEN rules) performed by FISs are:

- Compare the input variables with the membership functions on the antecedent part to obtain the membership values of each linguistic label. (this step is often called *fuzzification*.)

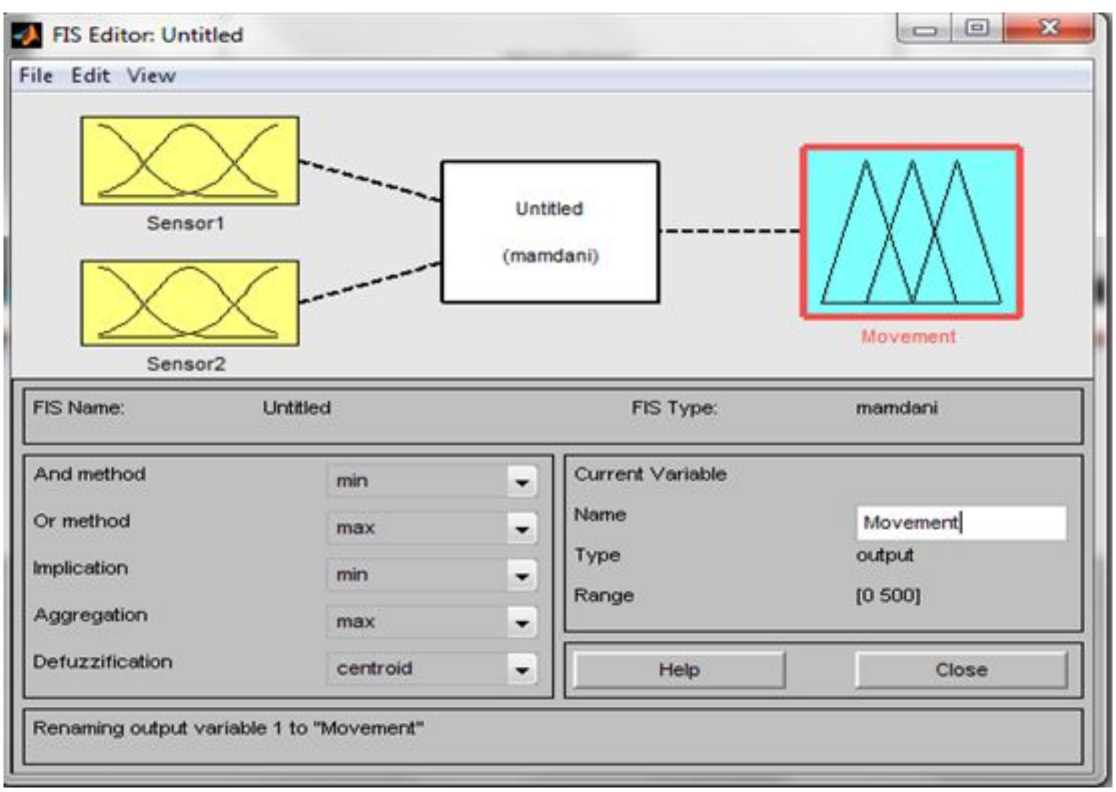
- Combine (through a specific *t*-norm operator, usually multiplication or min) the membership values on the premise part to get *firing strength* (*weight*) of each rule.

- Generate the qualified consequent (either fuzzy or crisp) or each rule depending on the firing strength.

- Aggregate the qualified consequent to produce a crisp output. (This step is called *defuzzification*.)

3.4.1.2 Fuzzification of Input

It is the block inside the controller which converts each piece of input data to degrees of membership by a looking up in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable. Fuzzification is an important concept in the fuzzy logic theory. Fuzzification is the process where the crisp quantities are converted to fuzzy (crisp to fuzzy). By identifying some of the uncertainties present in the crisp values, we form the fuzzy values. The conversion of fuzzy values is represented by the membership functions. In any practical applications, in industries, etc., measurement of voltage, current, temperature, etc., there might be a negligible error. This causes imprecision in the data. This imprecision can be represented by the membership functions. Hence fuzzification is performed. Thus fuzzification process may involve assigning membership values for the given crisp quantities.



3.4.2 Fuzzy Membership Functions for Outputs

Every element in the universe of discourse is a member of a fuzzy set to some grade, maybe even zero. The grade of membership for all its members describes a fuzzy set, such as Neg. In fuzzy sets elements are assigned a grade of membership, such that the transition from membership to non-membership is gradual rather than abrupt. The set of elements that have a non-zero membership is called the support of the fuzzy set. The function that ties a number to each element x of the universe is called the *Membership function u(x).*

The designer is inevitably faced with the question of how to build the term sets. There are two specific questions to consider:

(i) How does one determine the shape of the sets?

(ii) How many sets are necessary and sufficient?

3.4.2.1The Membership Function Editor

It is the tool that lets you display and edit all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system.

It features include

(1) Core

If the region of universe is characterized by full membership (1) in the set $A\sim$then this gives the core of the

membership function of fuzzy at $A\sim$.The elements, which have the membership function as 1, are the elements

of the core, i.e., here $\mu A\sim (x) = 1$.

(2) Support

If the region of universe is characterized by nonzero membership in the set $A\sim$, this defines the support of a

membership function for fuzzy set $A\sim$. The support has the elements whose membership is greater than 0. $\mu A\sim$

$(x) > 0$.

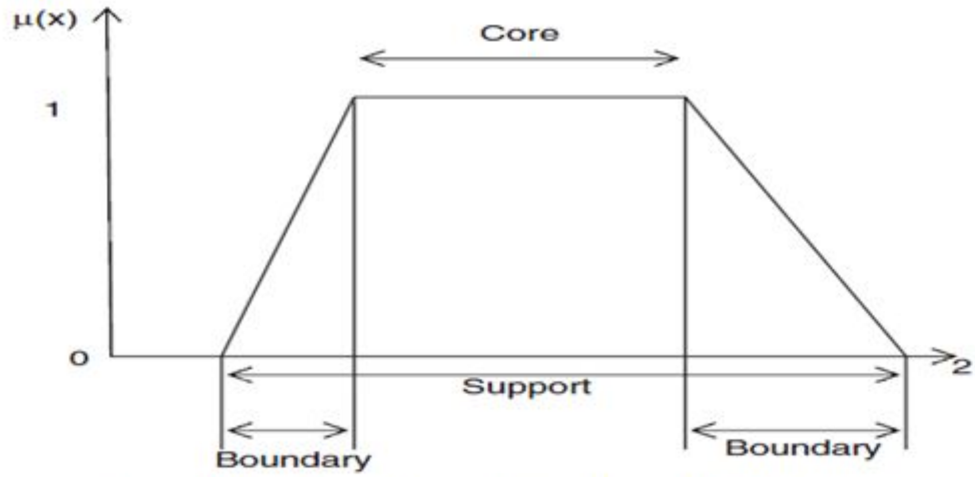


Fig. 4.1. Features of membership function

(3) Boundary

If the region of universe has a nonzero membership but not full membership, this defines the boundary of a membership; this defines the boundary of a membership function for fuzzy set $A\sim$. The boundary has the elements whose membership is between 0 and 1, $0 < \mu A\sim(x) < 1$.

These are the standard regions defined in the membership functions. Defining two important terms.

Crossover point

The crossover point of a membership function is the elements in universe whose membership value is equal to 0.5, $\mu A\sim(x) = 0.5$.

Height

The height of the fuzzy set $A\sim$ is the maximum value of the membership function,

FIGURE THE MEMBERSHIP FUNCTION FOR THE FIRST SENSOR

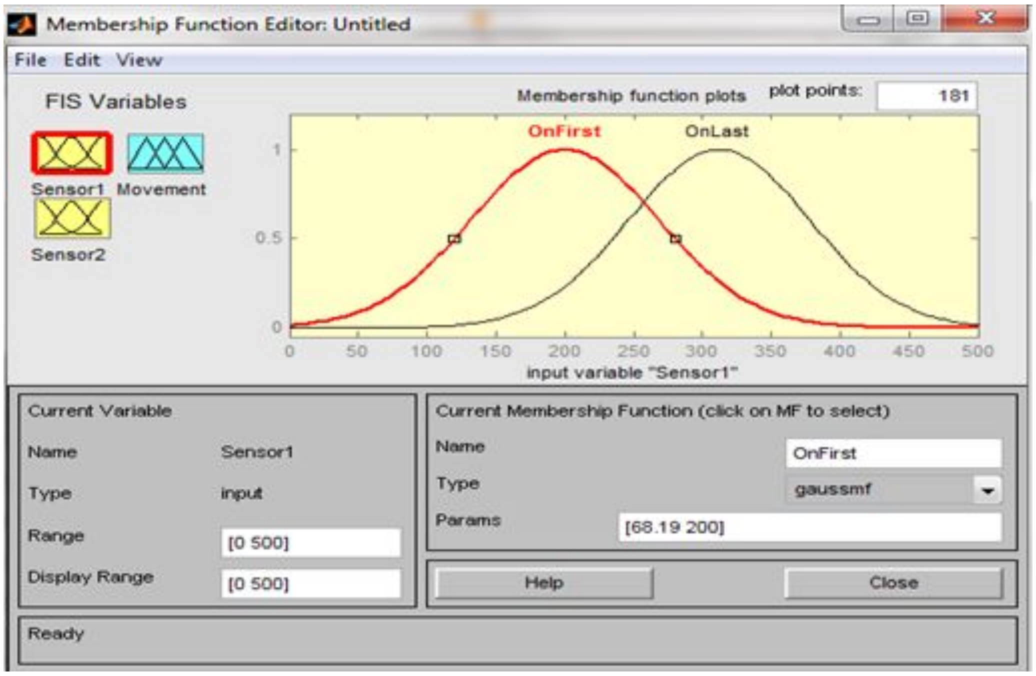


FIGURE THE MEMBERSHIP FUNCTION FOR THE SECOND SENSOR

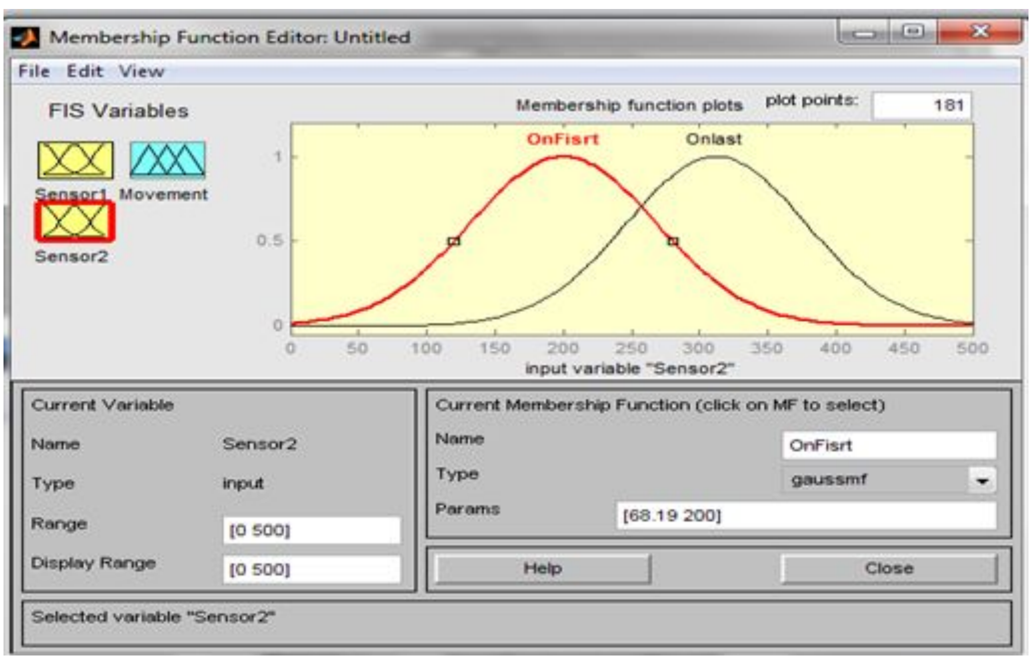
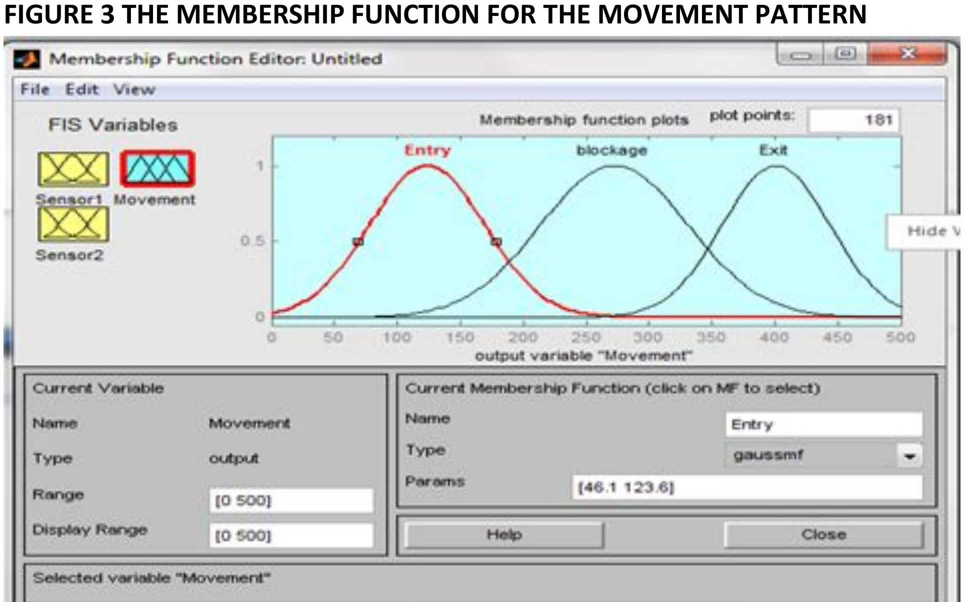


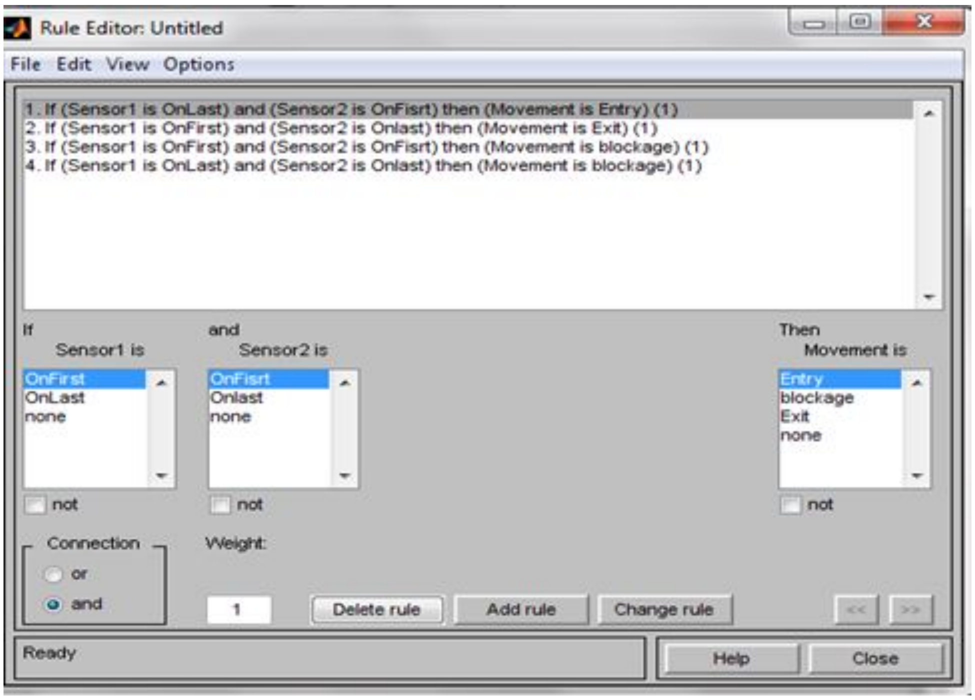
FIGURE 3 THE MEMBERSHIP FUNCTION FOR THE MOVEMENT PATTERN



3.4.3 Rule Base

The fuzzy rules are a collection of linguistic statements that describe how the FIS should make a decision regarding classifying an input or controlling an output. They are written in the following format:

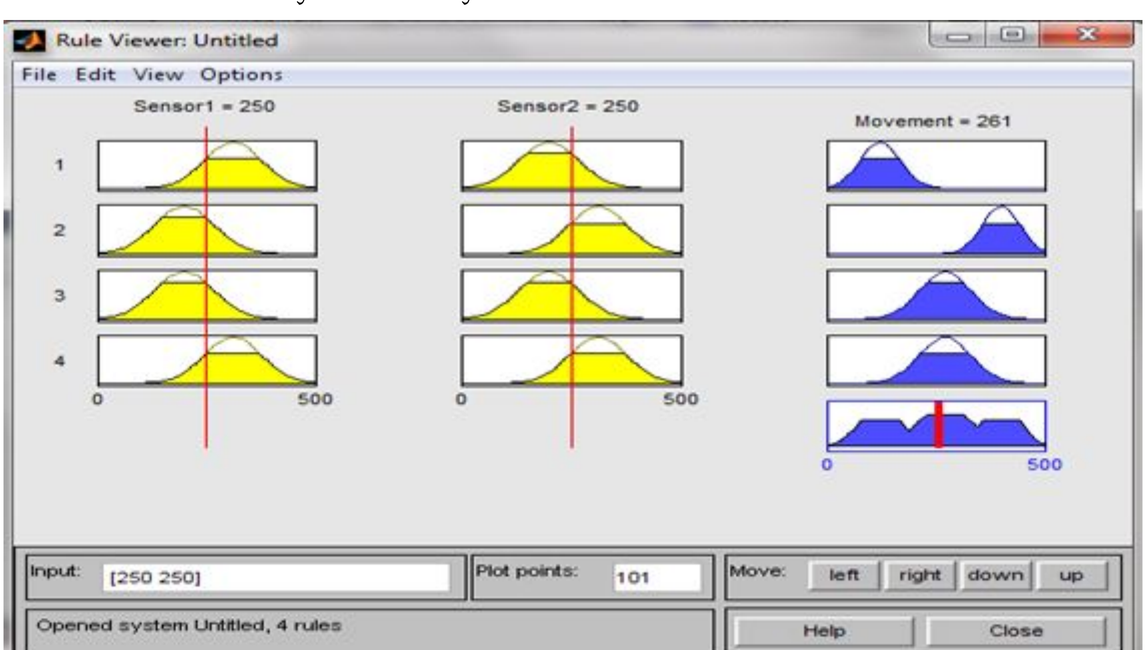
if *(input 1 is membership function 1)* ***and/or*** *(input 2 is membership function 2)* ***and/or***. . . ***then*** *(output is output membership function).*



3.4.5 THE RULE VIEWER

It allows you to interpret the entire fuzzy inference process at once. It also shows how the shape of certain membership functions influences the overall result. It plots every part of every rule, so it can become unwieldy for particularly large systems, but, for a relatively small number of inputs and outputs, it performs well .

The Rule Viewer shows one calculation at a time and in great detail hence in this sense, it presents a sort of micro view of the fuzzy inference system.



Defuzzification

It is a mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of non fuzzy (crisp) control action. This process is necessary because in many practical applications crisp control action is required to actuate the plant. The defuzzifier also performs an output denormalization if a normalization is performed in the fuzzification module.

There are seven methods used for defuzzifying the fuzzy output functions including

(1) Max-membership principle,

(2) Centroid method,

(3) Weighted average method,

(4) Mean–max membership,

(5) Centre of sums,

(6) Centre of largest area, and

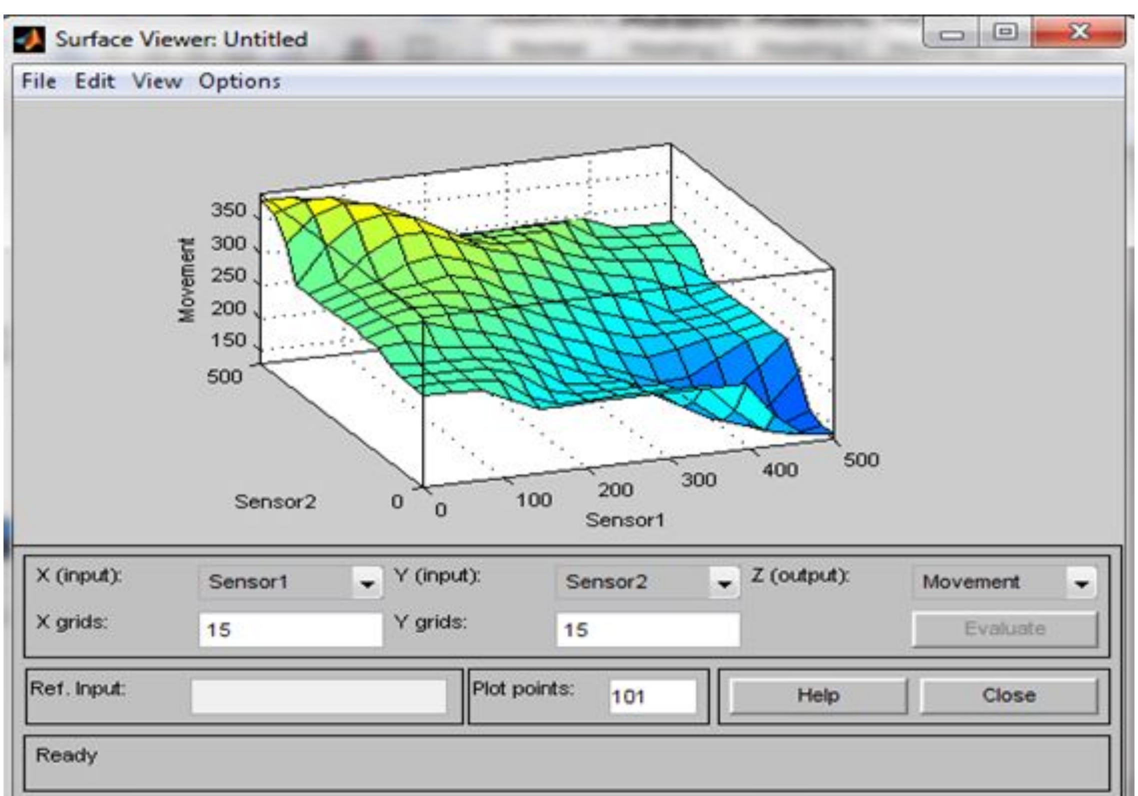
(7) First of maxima or last of maxima

3.4.6 The Surface Viewer

It helps you see the entire output surface of your system (the entire span of the output set based on the entire span of the input set) . This viewer is the last of the five basic Fuzzy Logic Toolbox GUI tools. The Surface Viewer have a three-dimensional curve that represents the mapping from sensor1 and sensor2 to the movement. Since the curve represents a two-input one-output case, you can see the entire mapping in one plot.

The Surface Viewer also has a special capability that is very helpful in cases with two (or more) inputs and one output: you can grab the axes, using the mouse and reposition them to get a different three-dimensional view on the data.

FIGURE THE SURFACE VIEWER OF THE TWO SENSORS Vs MOVEMENT



3.4 THE MICROCONROLLER

3.4.1 Power Supply

The FRDM-KL25Z offers a design with multiple power supply options. It can be powered from

- ✓ the USB connectors
- ✓ the V_{IN} pin on the I/O header
- ✓ an on-board coin cell battery
- ✓ or an off-board 1.71-3.6V supply from the 3.3V pin on the I/O header.

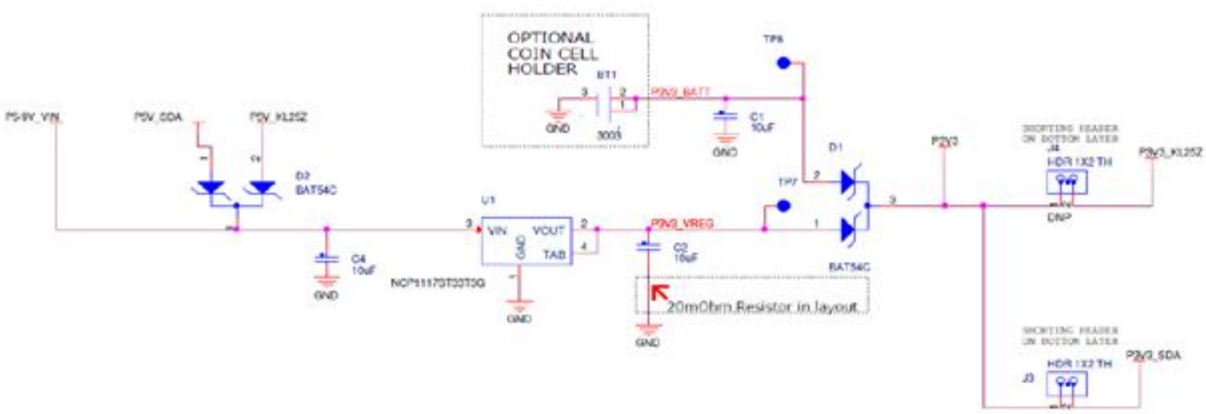
The USB and V_{IN} supplies are regulated on-board using a 3.3V linear regulator to produce the main power supply. The other two sources are not regulated on-board. Table 1 provides the operational details and requirements for the power supplies.

Table 1. Power Supply Requirements

Supply Source	Valid Range	OpenSDA Operational?	Regulated on-board?
OpenSDA USB (J7)	5V	Yes	Yes
KL25Z USB (J5)	5V	No	Yes
V_{IN} Pin	4.3-9V	No	Yes
3.3V Pin	1.71-3.6V	No	No
Coin Cell Battery	1.71-3.6V	No	No

Note that the OpenSDA circuit is only operational when a USB cable is connected and supplying power to J7. However, protection circuitry is in place to allow multiple sources to be powered at once.

THE CIRCUIT DIAGRAM OF THE POWER SUPPLY



CHAPTER 4 THE IMPLEMENTATION PROCESS

The chapter involves a discussion of the implementation of the algorithms in the CooCox Coide software using C/C++ programming language to program a sensor pattern. The Matlab display is used to display the sensor conditions at any point while the seven segment display is used to displays the number of people in the room.

4.1 System Overview

The architecture of the system consist of the following elements.

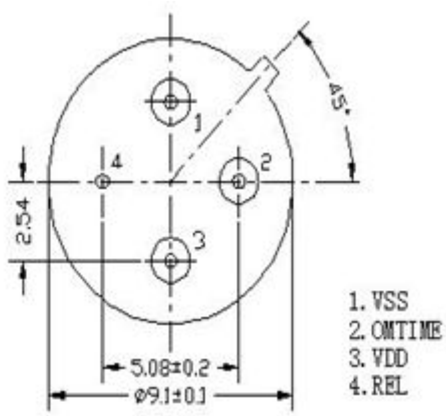
The People Counting Device. It uses a double element passive infrared sensor (PIR) connected to a microcontroller. The microcontroller continuously analyses the PIR's output signal to detect and count the number of people passing it. A successful pass is a traversal from one side of the PIR's field of view to the other. Each pass increases or decreases the count which is then displayed on the Seven segment LED display.

4.2 Hardware Configuration

For custom-development of the monitoring and counting device, the freescale microcontroller (FRDM-KL25Z)was used since it provides a configurable and programmable processor board which can be easily combined with the quality sensors .The sensor's are evaluated for the 48MHz-operated processor. A PIR sensor and a UART module for communication to the sensor terminal. Both the processor board and the UART module support low-power modes, which we make no use of in this prototype scenario.

4.2.1 THE AM 412 PIR SENSOR

THE SENSOR PIN LAYOUT AND TECNICAL CONNECTION DATA



■ Technical Data

1. Maximum Ratings

Characteristics	Symbol	Min. Value	Max. Value	Unit	Remarks
Supply Voltage	V_{DD}	2.7	3.3	V	
Working Temperature	T_{ST}	-20	85	℃	
Current into any pin	Into	-100	100	mA	
Storage Temperature	T_{ST}	-40	125	℃	

2.Working Conditions (T=25°C, Vdd=3V, Except other requirements)

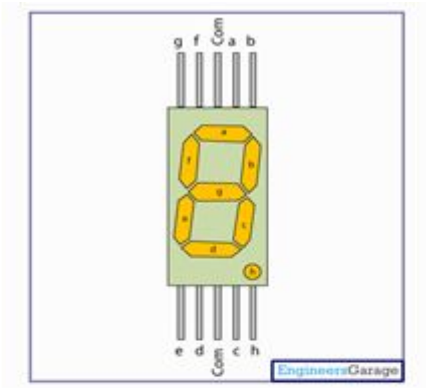
Characteristics	Symbol	Min.	Type	Max.	Unit	Remarks
Supply Voltage	V_{DD}	2.7		3.3	V	I_R=0.5mA
Working Current	I_{DD}		12	15	μA	
Output REL						
Output Low Current	I_{OL}	10			mA	V_{OL}<1V
Output High Current	I_{OH}			-10	mA	V_{OL}>(V_{DD}-1V)
Input ONTIME						
Voltage Input Range		0		V_{DD}	V	0V to ¼ V_{DD}
Input Bias Current		-1		1	μA	

4.2.2 THE SEVEN SEGMENT LED DISPLAY

It is the most basic electronic display device that can display digits from 0-9. it's configuration has an array of eight LEDs arranged in a special pattern to display these digits. They are laid out as a squared-off figure '8'. Every LED is assigned a name from 'a' to 'h' and is identified by its name. Seven LEDs 'a' to 'g' are used to display the numerals while eighth LED 'h' is used to display the dot/decimal.

It is generally available in ten pin package. While eight pins correspond to the eight LEDs, the remaining two pins (at middle) are common and internally shorted. These segments come in two configurations, namely, Common cathode (CC) and Common anode (CA). In CC configuration, the negative terminals of all LEDs are connected to the common pins. The common is connected to ground and a particular LED glows when its corresponding pin is given high. In CA arrangement, the common pin is given a high logic and the LED pins are given low to display a number.

LED PIN	PORT CONNECTED
a	PTE20
b	PTE21
c	PTE22
d	PTE23
e	PTE29
f	PTE13
g	PTD5
h	PTD0



The pattern display table

NUMBER TO DISPLAY	LED RESPOSIBLE	PORTS ENABLED ON FRDM-KL25Z
0	a,b,c,d,e,f	PTE20,PTE21,PTE22,PTE23,PTE29,PTA13
1	b,c	PTE21,PTE22
2	a,b,g,e,d	PTE20,PTE21,PTE23,PTE29,PTD5
3	a,b,g,c,d	PTE20,PTE21,PTD5,PTE22,PTE23
4	f,g,b,c	PTA13,PTD5, PTE21,PTE22
5	a,f,g,c,d	PTE20,PTA13,PTD5,PTE22,PTE23
6	a,f,g,c,d,e	PTE20,PTA13,PTD5,PTE22,PTE23,PTE29
7	a,b,c	PTE20,PTE21,PTE22
8	a,b,c,d,e,f,g	PTE20,PTE21,PTE22,PTE23,PTE29,PTA13
9	a,b,f,g,c	PTE20,PTE21,PTA13,PTD5,PTE22

4.3 Software Configuration

In programming the prototype monitoring and counting device, algorithms were transformed and implemented into the C/C++ language. This is the language which was used to program the COOCOX COIDE software. The MATLAB commands were then used to draw the real time graphs of different response from the two sensors simultaneously. The programming skills used were UART,ADC,GPIO, implementing the algorithm.

4.3.1 Programming for UART

This is very essential to display the output on the screen. The output are transmitted from the microcontroller to the terminal of the computer for display on the computer screen, and also to be used by the MATLAB software command for plotting the sensors response graphs.

4.3.2 Programming for ADC (Analog to Digital Conversion)

The output of the sensors are analog signal. They are converted to their equivalent digital signal in 8 bits.

4.3.3 Programming for GPIO (General Purpose Input Output)

For different display of numbers on the Seven Segment LED Display, the responsible pins have to be set high and low depending on the display depending on the display to be achieved. The GPIO of FDRM KL25z was used to achieve this functionality

4.3.4 Implementing the count algorithm

It involved programming the algorithm

4.4 Limiting the Sensor's Field of View

The PIR sensor uses Fresnel lenses to focus the incident IR radiation, such that the sensor is enabled to cover a wider area and to detect movement in both the horizontal and vertical axes. Since we are only interested in signals within a specific horizontal range, let us restrict the sensor's coverage by applying a blocking material to the respective areas of the Fresnel lenses. This effectively limits the covered area by the sensor and thus reduces the risk of false counts.

CHAPTER 5 RESULTS AND EVALUATION

5.1 INTRODUCTION

To evaluate the algorithm and the monitoring and counting device performance, two distinct series of experiments were conducted. The first series was used to capture sensor data resulting from several different movement patterns, including single-person passages and multiple people passing in line. For evaluation, the sensor output was subsequently processed by the
C/C++ implementations of the COUNT algorithm.

All experiments were timed in order to gauge the effect of higher rates of passages. The same experiments were conducted using different timing parameters in order to simulate people walking up-close or at larger distances from one person to the next. The evaluation was conducted in an lab room, where the device observed the room's door to an adjacent hallway.

5.2 COUNT ALGORITHM PERFORMANCE

The following experiments were conducted for each of the time distances. The time distances give an approximate indication of how close to each other the participants walked in line. The speed at which to walk was not regulated,however, the lower the distance in between passing events, the quicker each preceding individual would make space for the next.

It checks for the four expected states then increments or decrements a counter based on the pattern detected.The chosen algorithm works best for single motion in any direction within the sensors field of view. It seems challenging when two or more people are passing at the same time at the door. This algorithm can only work for single movements e.i one movement at a time(entry or exit)

5.3 THE MATLAB DISPLAY

It displays the raw conditions from the sensors in real time.

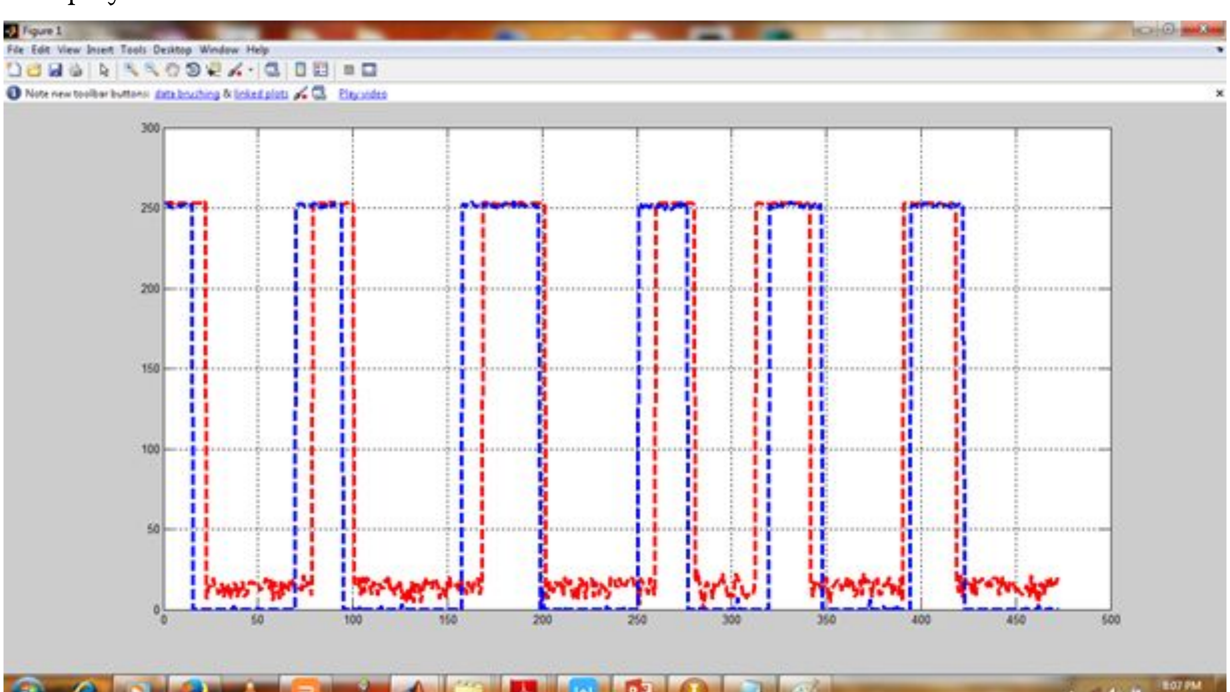


Figure 3.0 The raw sensor display in MATLAB

As the level of infrared is being detected by the sensors, the corresponding digital output values are sent to the terminal of the computer. This values are then taken by MATLAB as input, so it draws them in real time. This helps for easy visibility of the response from the sensors.

Thus the figure can be analyzed that it represent three people entering the room at 100, 200 , 275 seconds then two people left the room at 320 and 390 seconds from the initialization of the program

5.4 The seven segment LED display

It displays the updated number of people count. It shows the number of people in the room at that given moment since it is updated by the entries and exits on the door.

This makes it even simpler since you do not need the computer and MATLAB software to know the number of people in the room. Once the program is downloaded to the microcontroller (MKL25Z128VLK4) . The setup can be installed then this seven segment LED will display the number of people in the room at any given moment.

5.5 ACHIEVEMENTS

The project has enabled a clear understanding in the following sections.

1. The principal of operation of the all motion detectors or motion sensors. The project uses the AM412 PIR sensors.

2. The concept of fuzzy logic and how its used in simulation.

3. Coming up with a suitable counting algorithm

4. Applying both the concept of fuzzy logic and the developed algorithm in building a room occupancy monitoring and counting devise. The programs were written in COOCOX COIDE and MATLAB software. The hardware composed of the microcontroller FRDM-KL25Z and the seven segment LED display.

5.6 CHALLENGES

1. Intergrating MATLAB screen to display both the graphical values from the sensors and the update number of people in the room. This is why the seven segment LED display was used separately.

2. Obtaining the entry or exit time. This is because the internet was the only source of reference for the FRDM KL25Z board has very minimum discussion on some functionality of the board like setting timer

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

In some scenarios the obtained total occupancy for some observed space may be within acceptable limits of accuracy, despite each sensor's values.The aim of this project was to develop a room monitoring and counting devise. The Am412-PIR sensors were chosen for their simplicity and effectiveness among other advantages like Digital signal processing (DSP), Power adjustable, save more energy, Two-way differential high impedance sensor input, Built-in filter, screen the interference by other frequency, Excellent power supply rejection, Insensitive to RF interference Schmidt REL output, Low voltage, low power consumption, instantaneous settling after power up. In contrast to other approaches used for people counting, e.g. cameras or active infrared arrays,

I have considered the people counting problem from different perspectives. This included the development of algorithms to parse Am412-PIR sensor signals. The development of the algorithm in COOCOX COIDE was chosen due to the following advantages free to use, its a fully functional IDE, its component-oriented development platform, its internet based, efficient integration of network resources, it integrates CoOS and peripheral registers, and has a flash loader for Flash memory programming .The collection of signals from the sensor network is done by the microcontroller(KL25Z128VLK) and this was very possible due to the following advantages of the board a max operating frequency of 48MHz, 128KB of flash, a full-speed USB controller, and loads of analog and digital peripherals and the display of the room occupancy in seven segment LED was successful. The algorithms developed apply a novel approach to the people counting problem.

My overall experience in building a room occupancy monitoring and counting device was a great learning opportunity. I made several mistakes in the beginning but i was able to correct them. Overally, as far as enhancement is concerned, it was a great learning experience and this project has helped me to successfully build the room occupancy and counting device which when was tested yielded successful results.

6.2 Recommendation

While the evaluation demonstrates the adaptability of the people counting algorithm to various movement patterns, it reveals several open problems. In a real-life setting, people cannot be expected to pass an observed passage at presumed or timed intervals. In consequence, the COUNT algorithms should be able to detect several people passing at the same time through the door. A better display can be integrated for both the graphs and counted display. The system is recommended in monitoring movements in building for security reasons.

6.3 Appendix

1. Financial Budget

The budget for development of the prototype was Kshs 5899. The amount can be breaked down into the following expenditure.

Item	Quantity	Unit Price (Kshs)	Total (Kshs)
Microcontroller (FRDM – KL25Z)	1	3000	3000
Am 412 PIR Sensor	2	400	800
Terminal Display	1	950	950
Seven Segment LED Display	1	450	450
Resistors	7	7	49
Connecting wire	1 bundle	400	400
PCB board	1	250	250
Total			5899

2.Time Budget

From	To	Duration	Item	Scope

3. The Matlab Program

```
>> delete(instrfind);
instrfind
s = serial('COM11','BaudRate',56000)
fopen(s)
title('Sensors ADC Graphs');
xlabel('Time in seconds');
ylabel('Digital Value');
x=0;
while(1)
   data=fgets(s)
   if(strcmp(data,'')==0)
              x=x+1
         r = sscanf(data,'%d,%d'',')
         red(x) = r(1)
              blue(x) = r(2)
               plot(red,'r--','linewidth',3)
         drawnow
              grid on;
         hold on;
```

```
            plot(blue,'b--','linewidth',3)
            drawnow
        hold off;
    end
End
```

4. The COOCOX COIDE program

```
/**
 *****************************************************************************
 * @title   uart_freedomKL25.c
 * @author  0xc0170
 * @date    09 Dec 2012
 * @brief   UART example for Kinetis Freedom KL25 board. Baudrate is not gen
 *          erated in last driver for KL25. Check forum for more information
 *           or github to get latest drivers.
 *******************************************************************************
 */
////// The above comment is automatically generated by CoIDE //////////////////
/**
 *****************************************************************************
 * @title   uart_freedomKL25.c
 * @author  0xc0170
 * @date    30 Nov 2012
 * @brief   Print welcome message to the terminal through UART on
 *                KL25 Freedom board and echo users input typed in the terminal.
 *******************************************************************************
 */
#include "xhw_types.h"
#include "xhw_memmap.h"
#include "xdebug.h"
#include "xsysctl.h"
#include "xhw_sysctl.h"
#include "xhw_gpio.h"
#include "xgpio.h"
#include "xhw_uart.h"
#include "xuart.h"
#include "xhw_adc.h"
#include "xadc.h"
#include "xhw_ints.h"
#include "xcore.h"
#include "stdint.h"
#include "xhw_nvic.h"

#include "xgpio.h"
#include "xrtc.h"
#include "xhw_rtc.h"
/*
tTime  tTime1;
unsigned long ulj = 0;
unsigned long ulTimeAlarm[2] = {RTC_TIME_CURRENT, RTC_TIME_ALARM};

unsigned long xRTCSCallback(void *pvCBData, unsigned long ulEvent,
                            unsigned long ulMsgParam, void *pvMsgData)
{
    ulj++;
    return 0;
}
```

```c
void RTCTickInt()
{

    RTCTimeInit();

    tTime1.ulSecond = 40;

    tTime1.ulMinute = 20;

    tTime1.ulHour = 17;

    tTime1.ulMDay = 11;

    tTime1.ulMonth = 8;

    tTime1.ulYear = 2011;

    tTime1.ulWDay = 4;

    tTime1.ulFormat = RTC_TIME_24H;

    //
    // Writes current time to corresponding register.
    //
    RTCTimeWrite(&tTime1, ulTimeAlarm[0]);

    RTCIntCallbackInit(RTC_INT_TIME_TICK, xRTCSCallback);

    //
    // Enables tick interrupt.
    //
    RTCIntEnable(RTC_INT_TIME_TICK);

    xIntEnable(INT_RTCS);
    xIntMasterEnable();
      RTCStart();

    //while(ulj < 1);
    //xIntMasterDisable();
 }
*/
static volatile uint32_t _millisCounter;
unsigned long ulValueLength;
unsigned long ulData;
xtBoolean ADCmeasured;
xtBoolean ReturnValue;
xtBoolean Flag;
xtBoolean sensor1_activated = xfalse;
xtBoolean sensor2_activated = xfalse;
xtBoolean People_count_Flag = xfalse;

int people_count = 0;

const unsigned char Begin[] =
             "Welcome, here is Freedom KL25 UART example\r\n, \
                               Now you can input your word, end with 'enter'\r\n";
const unsigned char End[] = "\nEnd of UART Example";
const unsigned char adc_header[] = "The read ADC value = \0";
unsigned long ADCCallback(void *pvCBData, unsigned long ulEvent,
             unsigned long ulMsgParam, void *pvMsgData);

//The funtion reads from channel 15 of the adc register which corresponds to PTC1 of the Freescale
Board
//it then delays for one second then returns the value
unsigned long read_channel1() {
```

```
        ADCSingleEndedChannelEnable(xADC0_BASE, ADC_CHANNEL_15);
        SysCtlDelay(1000000);
        return ulData;

}
//The funtion reads from channel 14 of the adc register which corresponds to PTC0 of the Freescale
Board
//it then delays for one second then returns the value
unsigned long read_channel2() {
        ADCSingleEndedChannelEnable(xADC0_BASE, ADC_CHANNEL_14);
        SysCtlDelay(1000000);
        return ulData;
        //SysCtlDelay(1000000);
}


void SysTickIntHandler(void) {
_millisCounter++;
}
uint32_t millis(void) {
return _millisCounter;
}
void millisDelay(uint32_t DelayTime) {
static uint32_t delayTimer = 0;
delayTimer = millis();
while (millis() - delayTimer < DelayTime);

}
/////////////////////////////////////CODING THE SEVEN SEGMENT
LED/////////////////////////////////////

void read_Zero() {


                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display zero
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 0 );
                    xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 0 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 1 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );

}
void read_One() {



                          xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 1 );  // Light the pins to
display ONE
                          xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                          xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                          xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 1 );
                          xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 1 );
                          xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 1 );
                          xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 1 );
                          xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );


}
void read_Two() {
```

```
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display TWO
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 1 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 0 );
                         xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 1 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 0 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );


}
void read_Three() {


                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display THREE
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 1 );
                         xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 1 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 0 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );


}
void read_Four() {


                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 1 );  // Light the pins to
display FOUR
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 1 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 1 );
                         xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 0 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 0 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );


}
void read_Five() {


                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display FIVE
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 1 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 1 );
                         xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 0 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 0 );
                         xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );
}
void read_Six() {

                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display SIX
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 1 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                         xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 0 );
```

```
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 0 );
                    xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 0 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 0 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );
}
void read_Seven() {


                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display SEVEN
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 1 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 1 );
                    xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 1 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 1 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );


}
void read_Eight() {

                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display EIGHT
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 0 );
                    xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 0 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 0);
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );

}
void read_Nine() {


                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_20, 0 );  // Light the pins to
display NINE
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_21, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_22, 0 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_23, 1 );
                    xGPIOPinWrite( xGPIO_PORTE_BASE, xGPIO_PIN_29, 1 );
                    xGPIOPinWrite( xGPIO_PORTA_BASE, xGPIO_PIN_13, 0 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_5, 0 );
                    xGPIOPinWrite( xGPIO_PORTD_BASE, xGPIO_PIN_0, 1 );

}

void read_Display(int count) {
     switch (count)
     {
     case 0:
          read_Zero();
          break;
     case 1:
          read_One();
               break;
     case 2:
          read_Two();
               break;
     case 3:
          read_Three();
               break;
```

```
	case 4:
		read_Four();
			break;
	case 5:
		read_Five();
			break;
	case 6:
		read_Six();
			break;
	case 7:
		read_Seven();
			break;
	case 8:
		read_Eight();
			break;
	case 9:
		read_Nine();
			break;
	default:
		break;
	}

}

///////////////////////////////////////END/////////////////////////////////////////////////////////////
//

int main(void) {
	char buff[100];
	int val;
	int val1;
	int decVal;
	int decVal1;
	int sensor1;
	int sensor2;
	int sensorsTate;
	int stateCount = 0;


	static volatile uint8_t previous_state=0;
	static volatile uint8_t Current_state;
	static volatile uint8_t combined_state;
	static volatile uint8_t states[4];

	xHWREG(SIM_COPC) = 0x00; //disable watchdog
	//
	// Disable processor interrupts
	//
	// ReturnValue = xIntMasterDisable();
	xSysCtlClockSet(48000000, xSYSCTL_OSC_MAIN | xSYSCTL_XTAL_8MHZ);
	SysCtlDelay(1000000);

	//
	// Enable GPIO and UART Clock
	//
	SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
	SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);

	//
	// Remap UART pin to GPIO Port UART0_RX --> PA2 UART0_TX --> PA1
	//
	xSPinTypeUART(UART0RX, PA1);
	xSPinTypeUART(UART0TX, PA2);

	//
	// Set UART clock
```

```
//
SysCtlPeripheralClockSourceSet(SYSCTL_PERIPH_UART0_S_MCGPLLCLK_2);

//
// Disable UART Receive/Transmit
//
UARTDisable(UART0_BASE, UART_TX | UART_RX);

//
// Configure UART Baud 115200
//
UARTConfigSet(UART0_BASE, 9600,
             UART_CONFIG_SAMPLE_RATE_12 | UART_CONFIG_WLEN_8
                          | UART_CONFIG_PAR_NONE | UART_CONFIG_STOP_1);

//
// Enable UART Receive and Transmit
//
//UARTEnable(UART0_BASE, UART_TX | UART_RX);
UARTEnable(UART0_BASE, UART_TX);

//set adc
xSysCtlPeripheralEnable(SYSCTL_PERIPH_ADC);
xSysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);

//
// Set clocks for ADC - bus clock without divider
//
xSysCtlPeripheralClockSourceSet(xSYSCTL_ADC0_MAIN, 0);
xSPinTypeADC(ADC15, PC1);
xSPinTypeADC(ADC14, PC0);
xADCConfigure(xADC0_BASE, xADC_MODE_SCAN_SINGLE_CYCLE,
             ADC_TRIGGER_PROCESSOR);

//
// Enable the ADC end of conversion interrupt
//
xADCIntEnable(xADC0_BASE, xADC_INT_END_CONVERSION);

//
// Install the callback interrupt
//
xADCIntCallbackInit(xADC0_BASE, ADCCallback);

//
// Enable the NVIC ADC interrupt
//
xIntEnable(xINT_ADC0);

//
// Enable interrupts
//
// RTCTickInt();
ReturnValue = xIntMasterEnable();
//
// Configure pin PC1 for ADC

// UARTCharPut(UART0_BASE, 'x');
// SysCtlDelay(1000000);

////////////////////////////////////////////////////////////////
//
// Enable PORTB, pins 18 and 19
xSysCtlPeripheralEnable(xSYSCTL_PERIPH_GPIOB);
xGPIOSPinTypeGPIOOutput(PB18);
xGPIOSPinTypeGPIOOutput(PB19);
```

```c
	// Enable PORTD, pin 1
	xSysCtlPeripheralEnable(xSYSCTL_PERIPH_GPIOD);
	xGPIOSPinTypeGPIOOutput(PD1);
	xGPIOSPinTypeGPIOOutput(PD5);
	xGPIOSPinTypeGPIOOutput(PD0);

	// Enable PORTE, pin 20,21,22,23,29
		xSysCtlPeripheralEnable(xSYSCTL_PERIPH_GPIOE);
		xGPIOSPinTypeGPIOOutput(PE20);
		xGPIOSPinTypeGPIOOutput(PE21);
		xGPIOSPinTypeGPIOOutput(PE22);
		xGPIOSPinTypeGPIOOutput(PE23);
		xGPIOSPinTypeGPIOOutput(PE29);

		// Enable PORTA, pin 13
			xSysCtlPeripheralEnable(xSYSCTL_PERIPH_GPIOA);
			xGPIOSPinTypeGPIOOutput(PA13);

	while (1) {
		/*read_Zero();
		SysCtlDelay(50000000);
		read_One();
		SysCtlDelay(50000000);
		read_Two();
		SysCtlDelay(50000000);
		read_Three();
		SysCtlDelay(50000000);
		read_Four();
		SysCtlDelay(50000000);
		read_Five();
		SysCtlDelay(50000000);
		read_Six();
		SysCtlDelay(50000000);
		read_Seven();
		SysCtlDelay(50000000);
		read_Eight();
		SysCtlDelay(50000000);
		read_Nine();
		SysCtlDelay(50000000);*/


		val = read_channel1();//attached to first sensor....sensor close to outdoor
		val1 = read_channel2();//attached to second sensor


		//It outputs the digital values from the two sensors


	/////////////////////////////////////////CAKES/////////////////////////////////////////////////
////
		//DECODING THE ADC CHANNEL VALUES INTO A 1 OR A 0
		if (val>200)
		{
			decVal = 1;
		}
		else
		{
			decVal = 0;
		}
		if (val1>200)
		{
			decVal1 = 1;
		}
		else
		{
```

```
				decVal1 = 0;
			}
			//if ((decVal != 0) || (decVal1 !=0))
			//{
				//SysCtlDelay(1000000);
			//}
		// STORE THE DECORDED VALUE IN ONE VARIABLE
			Current_state = (decVal<<0)|(decVal1<<1);
			//COMPARE WITH THE PREVIOUS STATE
			if (previous_state != Current_state)
			{


				previous_state = Current_state;
				states[stateCount] = Current_state;
				stateCount ++;
			}

			if (stateCount == 4)
			{
				//People_count_Flag = xtrue;
				combined_state = states[0] | (states[1] << 2) | (states[2] << 4)| (states[3] <<
6);
				if (combined_state == 0x2D) {//0x2d=00101101
					people_count ++;

				}
				if (combined_state == 0x1E){//0x1E=00011110
								people_count --;
				}

				combined_state = 0;
				stateCount = 0;
			}
			//if (previous_state != Current_state){
			   sprintf(buff, "%d,%d....%d..secs%d\n\r\0", val, val1,people_count);
								int j;
							   for (j = 0; j < strlen(buff); j++)
											{
											UARTCharPut(UART0_BASE, buff[j]);
											}
								read_Display(people_count);
								//xHWREG(SIM_COPC) = 0x00; //disable watchdog
							   SysCtlDelay(1000000);
				//}

		}
		return 0;
}

unsigned long ADCCallback(void *pvCBData, unsigned long ulEvent,
			unsigned long ulMsgParam, void *pvMsgData) {
		// Read data from result register
		ulData = xHWREG(xADC0_BASE + ADC0_RA);
		ADCmeasured = xtrue;

		return 0;
}
```