FPGA Based Dust Monitoring System

Group Project Report

Sl. No.	Reg. No.	Student Name	Department
1	15ETEC033002	Bharath Ranganath B K	VLSI
2	15ETEC033003	Dewang Shukla	VLSI
3	15ETEC033004	Gurubasavarajaiah N M	VLSI
4	15ETEC033006	M Tousif Ahmed	VLSI
5	15ETEC033008	Naresh Gowda M	VLSI
6	15ETEC033010	Prakruthi U S	VLSI
7	15ETEC033014	Shahmustafa Mujawar	VLSI

Mentor: Mr. Abdul Imran Rasheed

February – 2017

M.Tech. FT-2015

FACULTY OF ENGINEERING AND TECHNOLOGY

Ramaiah University of applied sciences

Bengaluru -560 054

FACULTY OF ENGINEERING AND TECHNOLOGY

FET

Certificate

*This is to certify that the Project titled "**FPGA Based Dust Monitoring System**" is a bonafide record of the group project work carried out by Bharath Ranganath B k, Dewang Shukla, Gurubasavarajaiah N M, M Tousif Ahmed, Naresh Gowda M, Prakruthi U S and Shahmustafa Mujawar bearing Reg. No. 15ETEC033002, 15ETEC033003, 15ETEC033004, 15ETEC033006, 15ETEC033008, 15ETEC033010, 15ETEC033014 Department of Electronics and Communication Engineering, FT-2015 batch in partial fulfilment of requirements for the award of M. Tech Degree of Ramaiah University of Applied Sciences.*

February – 2017

Mentor

Mr. Abdul Imran Rasheed

Dr. Hariharan Ramasangu
(HOD-ECE)

Prof. H. K. Narahari
Dean - FET

Declaration

FPGA Based Dust Monitoring System

The Group Project report submitted herewith is a result of our own work and in conformance to the guidelines against plagiarism as laid out in the University Student Handbook. All sections of the text and results which have been obtained from other sources are fully referenced. We understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Team members

Sl. No.	Reg. No.	Student Name	Department
1	Bharath Ranganath B k	15ETEC033002	VLSI
2	Dewang Shukla	15ETEC033003	VLSI
3	Gurubasavarajaiah N M	15ETEC033004	VLSI
4	M Tousif Ahmed	15ETEC033006	VLSI
5	Naresh Gowda M	15ETEC033008	VLSI
6	Prakruthi U S	15ETEC033010	VLSI
7	Shahmustafa Mujawar	15ETEC033014	VLSI

Date: **07 March 2017**

Contents

TEAM: Coming Together is Beginning, Keeping Together is Progress, and Working Together is Success

We express our sincere thanks to those individuals whose continuous support, direct and indirect contributions made this project possible.

We thank the management of **M. S. Ramaiah University of Applied Sciences**, Vice-chancellor **Dr. S. R. Shankapal** and Dean **Dr. H. K. Narahari** for all the facilities and encouragement.

We extend our thanks to **Mr**. **Abdul Imran Rasheed** who is the mentor of our project, without his help, guidance and source of information, it would not have been possible to bring out completion of our project.

Our sincere regards to all the staff members of Department of ECE for providing us with creative ideas and knowledge in the field of our project. We also extend our thanks to our **friends** and well-wishers for their timely help and support during the perusal of the project. Last but not the least our sincere thanks to **family members** for their constant prayers, cooperation and encouragement during the period, which served as a beacon light for us.

Abstract

Dust pollution comes primarily from the construction industry and related processes, like concrete crushing, cement batching and road stone plants, coal mines. Pollution caused by dust and particles poses the grave danger to people with heart or lung disease and other chronic diseases are at increased risk from dust and particle pollution. Studies have shown that when particles levels are high lead to intensify the disease. An IIT Kanpur study of Delhi's air last year says 38% of particulate matter (PM) 2.5 and 56% of PM 10 in the air is because of road dust.

The developed system has focused on checking dust present in the air. It is basically a Dust Density Concentration Measurement System that makes use of the Wi-Fi technology available to transmit real-time dust values of an environment to the HTTP server and which have been monitor by the web page and android application. A range of equipment with sophisticated data collection and analysis software has been supplied as part of a dust monitoring system. The optical air quality sensor used to sense dust particles in the environment. The MicroBlaze Processor-based architecture of interface control unit designed and developed with its peripherals to get access to data coming from the sensor and process it. The processed sensor data configured with the developed HTTP server and Wi-Fi module. The web page developed and configured with the HTTP server to show the dust density concentration obtained from the sensor. The micro SD card linked with the HTTP server to provide web page related data to the end-user. The android app has been configured with the HTTP server to show the dust density concentration on android mobile or tablet.

FPGA based dust monitoring system is successfully implemented on EDK and SDK platform. The entire FPGA based dust monitoring system is prototype on IoT platform by interfacing with web page and android device. The developed design has been prototyped on Nexys 4 Artix 7 FPGA board with optical dust sensor as sample. The developed system has focused on transmitting the real time dust values present in the environment around to the end user using available Wi-Fi technology.

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Introduction:

Air pollution is regarded as the pollution that has the greatest impact on the environment. Global warming phenomena that threaten our earth in recent years is closely related to the level of air pollution that caused by vast amount of carbon emission from vehicle petrol and diesel combustion, waste from ever-expanding production industry, uncontrolled combustion of fossil fuel and open-air burning in the waste management site. Hence, air pollution gradually becomes the focus of the world to address. Apart from the man-made causes, natural phenomena also contribute to air pollution, such as forest fires caused by heat from the sun and volcanic eruption capable bursting out a huge amount of particles, smoke and dust with large amounts of visual impairment in the vicinity of the incident.

Air pollution under the roof is also the growing problem nowadays. As individual spend 90% of their time in indoors (schools, offices, institutions, commercial facilities), Indoor Air Quality (IAQ) is the most influential factor for people health, comfort and safety. Indoor Air pollution causes the health problems known as Sick Building Syndrome (SBS) and Building Related Illness (BRI). The symptoms of SBS do not fit the pattern of any particular illness. It is very difficult to trace to any specific source, and relief from this symptoms occurs upon leaving the building. However, symptoms associated with Building Related Illness (BRI) are not relieved after leaving the building. BRI are caused by microbial contamination or specific chemical exposure that can result in allergy. The quality of indoor air depends upon a number of factors including the concentration of various gaseous and particulate pollutants. The air monitoring system for a smart home will be in basic requirement.

1.2 Air pollution Index (API):

The API is an index for reporting daily air quality. Air pollution index normally includes the major air pollutants that could hard to human health. API provides understandable information about the air pollution to the public. National Air Quality Index (NAQI) of India follows closely to Pollutant Standard Index (PSI) developed by United States Environmental Agency (EPA). The air pollutant included in India's API are carbon monoxide (CO), nitrogen dioxide (NO_2), sulphur dioxide (SO_2), ozone (O_3) and suspended particulate matter less than 10 microns in size (PM10).

Table 1.1: API System Adopted in India

Air Pollutant Index	Remark	Colors	Level of Health Concern
0 to 50	Good	Green	Minimal impact
51 to 100	Satisfactory	Light Green	Minor breathing discomfort to sensitive people
101 to 200	Moderate	Yellow	Breathing discomfort to the people with lungs, asthma and heart diseases
201 to 300	Poor	Orange	Breathing discomfort to most people on prolonged exposure
301 to 400	Very Poor	Red	Respiratory illness on prolonged exposure
401 to 500	Severe	maroon	Affects healthy people and seriously impacts those with existing diseases

1.2 Motivation:

Dust Monitor is a Dust Concentration Measurement System that makes use of the most modern technology available to transmit real time dust values of environment. A range of equipment with sophisticated data collection and analysis software can be supplied as part of a dust concentration monitoring network. These dust monitoring networks can range from basic systems monitoring dust concentrations in either PM2.5, PM10 or TSP t sophisticated systems measuring PM2.5, PM10 and TSP to US EPA certification (EQPM-0798-122) standards for automatic monitoring, recording and reporting. All equipment used in these Dust Monitoring Networks is robust and features low cost operations, ease of use and minimal maintenance. The networks can consist of mobile units, fixed stations or a combination of both.

Dusts are fine particles that present an explosion hazard when suspended in air under certain conditions. A dust explosion can cause catastrophic loss of life, injuries, and destruction of buildings. The U.S. Chemical Safety and Hazard Investigation Board (CSB) identified 281 combustible dust incidents between 1980 and 2005 that led to the deaths of 119 workers, injured 718, and extensively damaged numerous industrial facilities.

Benefits:

- Mobile units are self-contained and do not require any utility supply from the client, whilst the fixed stations generally would require a power supply

- All monitoring station can have various communications options installed, including radio networks, 3G (GSM), satellite or fixed line
- By making use of these dust monitoring networks, the client will be in position to know in real time what the dust concentration levels are in and around their operation, what the potential is for exceeding legal limitations and further be able to provide accurate real time reporting both internally and to external authorities when required, and as this data is all in real time, it will allow the client to pro-actively implement Dust Management Solutions to prevent localized dust concentration exceedance from becoming liability issues

CHAPTER 2

Background Theory

2.1 What is Particulate Matter (PM):

Particulate matter is defined by the United States Environment Protection Agency (EPA) has an air-suspended mixture of both solid and liquid particles. They are often separated into three classifications, called, coarse, fine and ultrafine particles. Coarse particles will have diameter of between 10µm and 2.5µm and settle relatively quickly whereas fine class of particulate matter will have diameter range of 0.1µm to 2.5µm and ultrafine will have, less than 0.1µm diameter particles and these ultrafine particles remain in suspension for longer period.

The PM10 refers to particles of diameter less than 10µm. these particles include dust, pollen and mold spores. Similarly, the PM2.5 refers to particles smaller than 2.5µm, the particles include combination particles, organic compounds and metals.

2.2 Sensor:

Here GP2Y1010AU0F is an optical air quality sensor, designed to sense dust particles. An infrared emitting diode and a phototransistor are diagonally arranged into this device, to allow it to detect the reflected light of dust in air. It is especially effective in detecting very fine particles like cigarette smoke, and is commonly used in air purifier systems.

Figure 2.1 GP2Y1010AUOF Dust Sensor

The sensor has a very low current consumption (20mA max, 11mA typical), and can be powered with up to 7VDC. The output of the sensor is an analog voltage proportional to the measured dust density, with a sensitivity of 0.5V/0.1mg/m^3.

Mating Connector Housing for Dust Sensor:

To interface with the sensor we need to connect to its 6-pin, 1.5mm pitch connector; we do have a mating connector for this. This is a 6-pin 1.5mm pitch connector which mates directly with the connector on the Sharp GP2Y1010AU0F optical dust sensor.

Figure 2. 2 Mating Connector Housing for Dust Sensor

2.2.1 Working of the Dust Sensor:

This dust sensor "GP2Y1010AU0F" is the device to detect house dust, cigarette smoke, etc. and designed as a sensor for automatic running of application like air purifier and air conditioner with air purifier function.

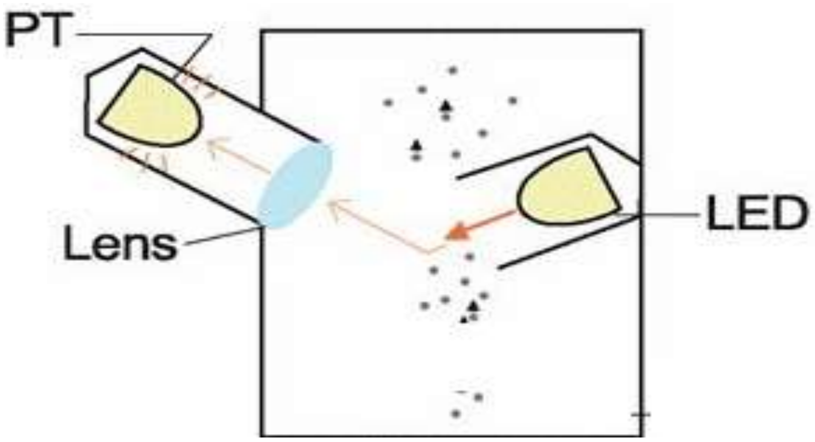


Figure 2. 3: Dust Sensor Working Principle

- This sensor has two basic elements, one is infrared emitting diode and photo detecting transistor
- These elements are arranged in diagonally into this device, to allow it to detect the reflected light from dust particles
- The device makes voltage output even when dust is not being detected
- This output voltage at no dust condition can specified as Vnd, this no dust voltage is due to reflected light from case of the device and its some part is get detected by phototransistor

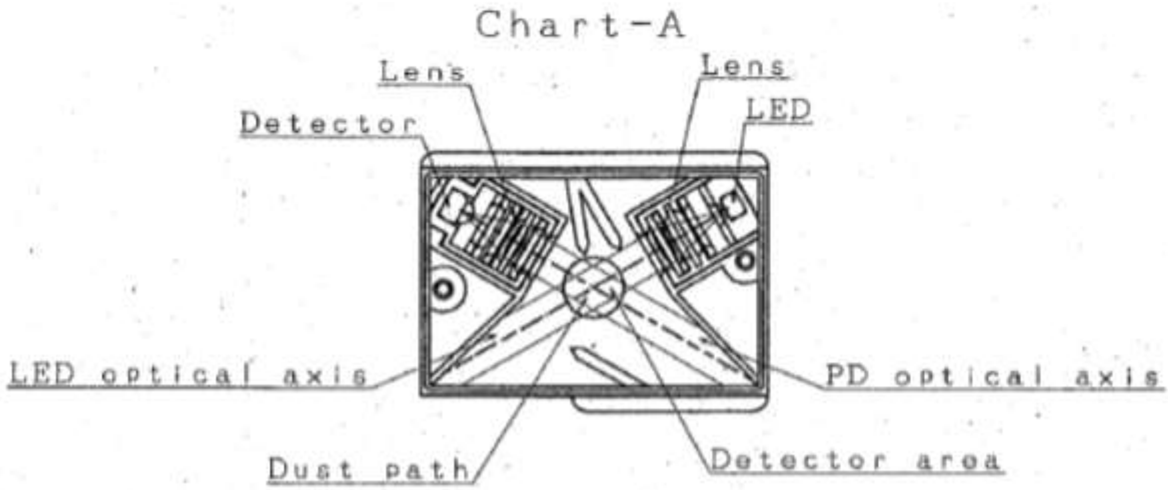


Figure 2. 4: Normal Condition

Light from the light emitter (Light Emitting Diode) is spotted with a lens and a slit as shown on the chart-A. Also for the light detector (Photodiode), a lens and a slit is positioned in front of it to cut disturbance light and to detect light reflection (when detecting dust) efficiently. Area where those two-optical axis cross is detection area of the device. Chart-B shows what is ongoing inside of the device when no dust exists and Chart-C shows that when dust exists.

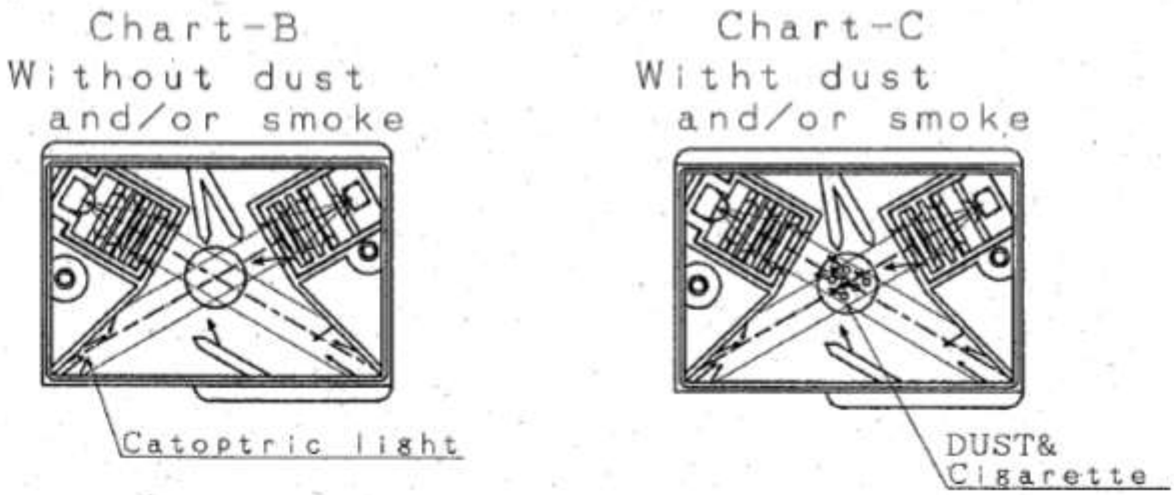


Figure 2. 5: B:No Dust Condition and A:Dust Condition

The device makes voltage output even when dust is not being detected. This output voltage at no dust condition is specified as Voc on the specification. This is because light emitted from the LED reflects at case of the device & some part of it gets to the detector. Chart-C shows how the device works when dust and/or cigarette smoke exists inside of it. In this case, the detector detects the light reflected from the dust and/or a particle of the cigarette smoke. Current in proportion to amount of the detected light comes out from the detector and the device makes analog voltage output (Pulse output) after the amplifier circuit amplifies the current from the detector.

2.3 Behavioural Model of Dust Sensor:

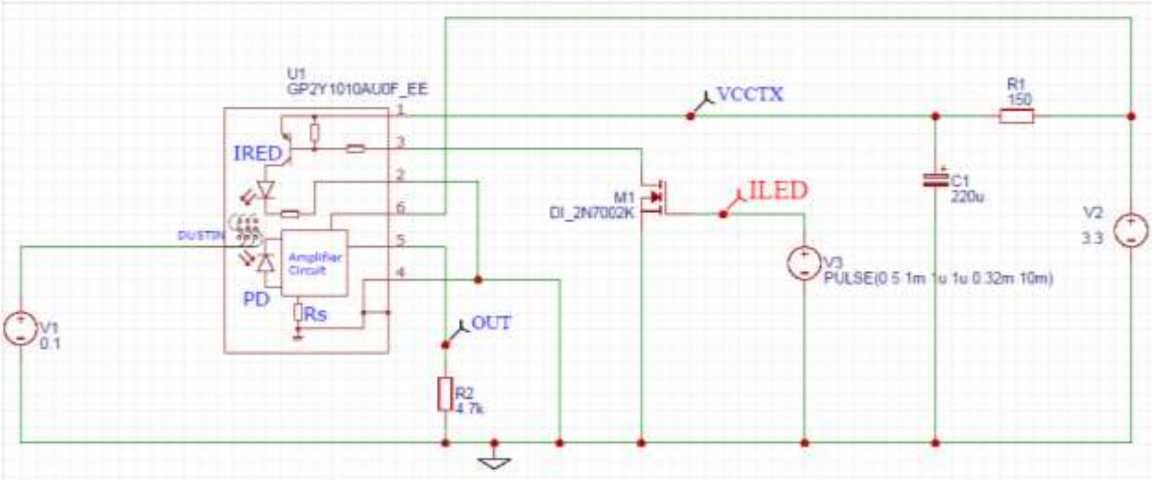


Figure 2. 6: Schematic of Behavioural circuit

2.4 Signal Conditioning Circuit:

Signal conditioning circuits are used to process the output signal from sensors of a measurement system to be suitable for the next stage of operation.

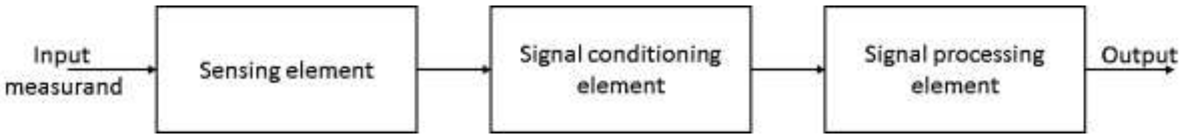


Figure 2.7: Elements of a measuring system

The success of the design of any measurement system depends on the design and performance of the signal conditioning circuits.

2.3.1 Internal schematic of dust sensor:

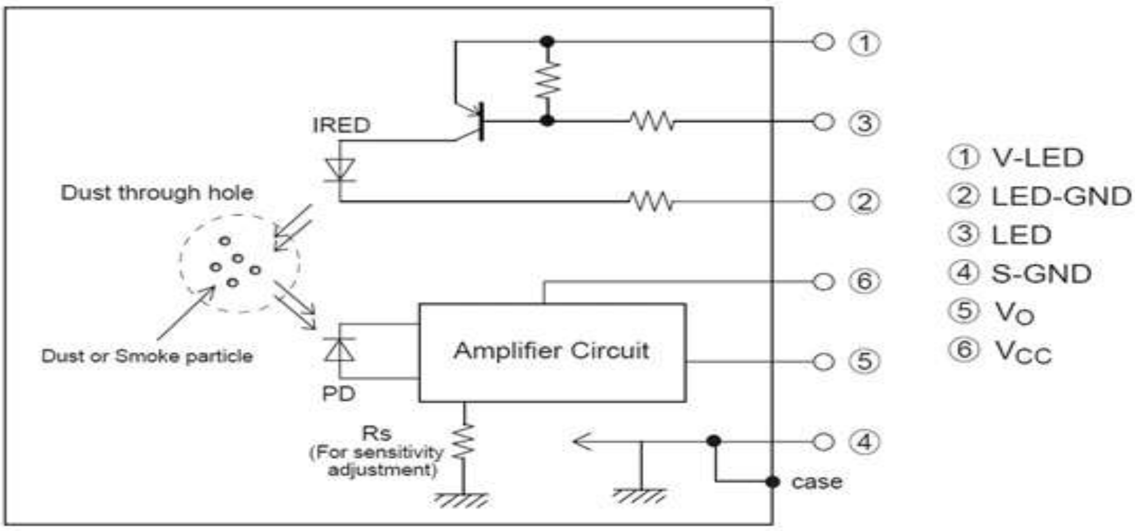


Figure 2. 8: Internal schematic of Dust Sensor

Signal conditioning contains blocks like amplification, filtering, range matching, isolation required to make sensor output suitable for processing after conditioning.

- Amplification: Signal amplification performs two important functions: increases the resolution of the input signal, and increases its signal-to-noise ratio
- Filtering: It is the most common signal conditioning function, as usually not all the signal frequency spectrum contains valid data[3]
- Isolation: Signal isolation must be used in order to pass the signal from the source to the measuring device without a physical connection

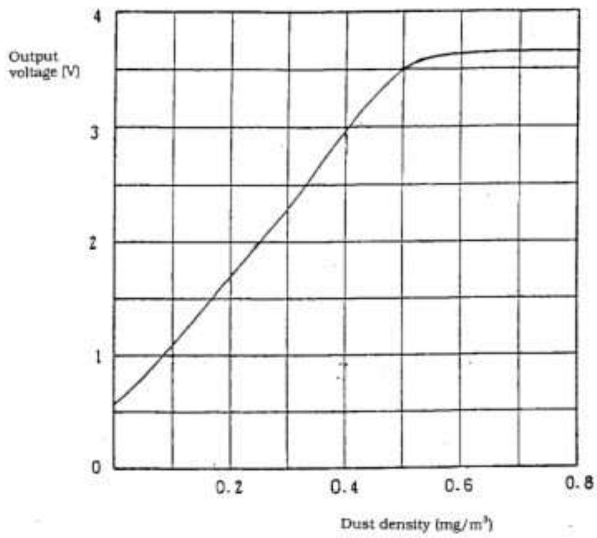


Figure 2. 9: Output Characteristics of Dust Sensor

Graph of the output voltage, compared to the total density of powder. This chart shows what you would get if you use the sensor as expected from SHARP. Instead the output signal from our adapter is not a voltage variable like this, but different impulses amplitude, Depending on the size of individual grains.

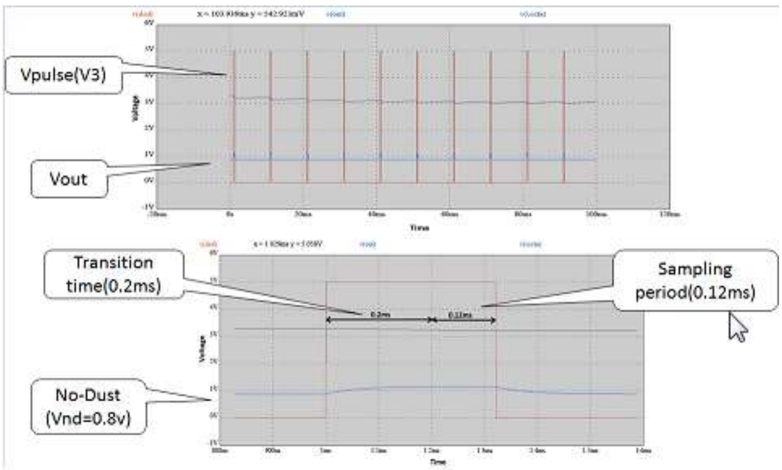


Figure 2. 10: Waveform of Behavioural Model

2.4 FPGA- Field Programmable Gate Array:

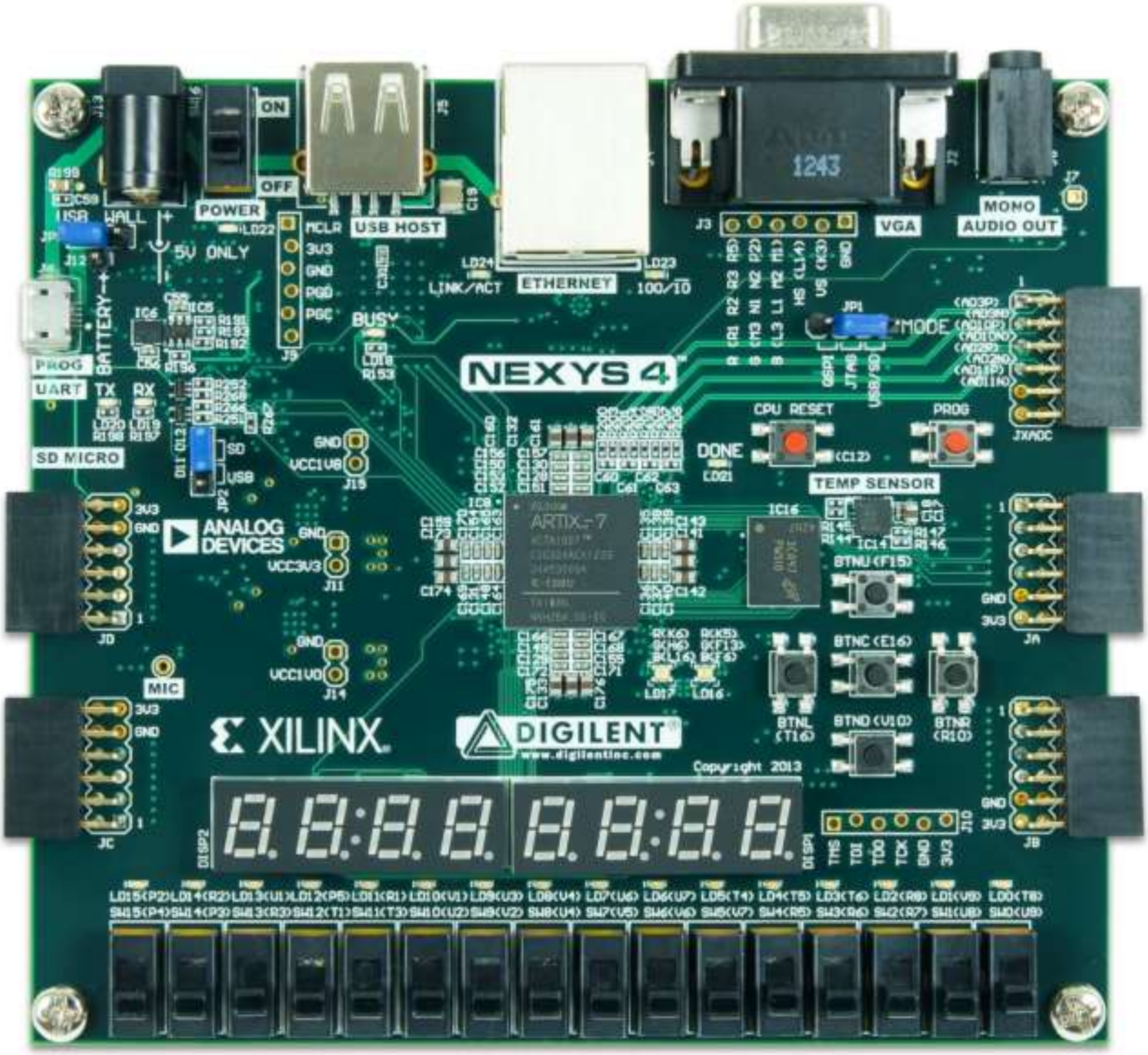


Figure 2. 11: Nexys-4 Artix-7 FPGA Board

2.4.1 Features:

- 10/100 Ethernet PHY
- 12-bit VGA output
- 15,850 logic slices, each with four 6-input LUTs and 8 flip-flops
- 16 user LEDs
- 16 user switches
- 240 DSP slices
- 3-axis accelerometer
- Internal clock speeds exceeding 450 MHz

- PDM microphone
- PWM audio output
- Temperature sensor
- Two 4-digit 7-segment displays
- Two tri-color LEDs
- USB-UART Bridge
- Xilinx Artix-7 FPGA XC7A100T-1CSG324C
- microSD card connector

2.4.2 Specifications:

- Company: Digilent
- Price: ₹ 42,985
- Size: 228.6 x 248.6 mm
- Xilinx Artix-7
  - Six clock management tiles, each with phase-locked loop (PLL)
  - 240 DSP slices
  - Internal clock speeds exceeding 450 MHz
  - On-chip analog-to-digital converter (XADC)
- Expansion Header
  - 2x6 Pmod ports
- Memory Devices
  - 128MB DDR2 SDRAM
- General User Input / Output
  - Two tricolor LED
  - 16-Slide switches
  - 6-Push buttons
  - 16-individual LEDs
  - 8-digit 7-segment display
- G-Sensor
  - Microphone
  - Temperature sensor
  - 3-Axis Accelerometer
- Power Supply:
  - 5-V form micro-USB port or from external power jack

- Interface
  - Micro SD card connector
  - Ethernet
  - USB to UART bridge
  - VGA Port

2.5 Digital PMOD Wi-Fi Module:

The Pmod WiFi provides Wi-Fi access through the Microchip® MRF24WG0MA Wi-Fi™ radio transceiver module. Users can communicate with the IEEE 802.11g compliant chip through SPI and achieve data rates up to 54 Mbps.

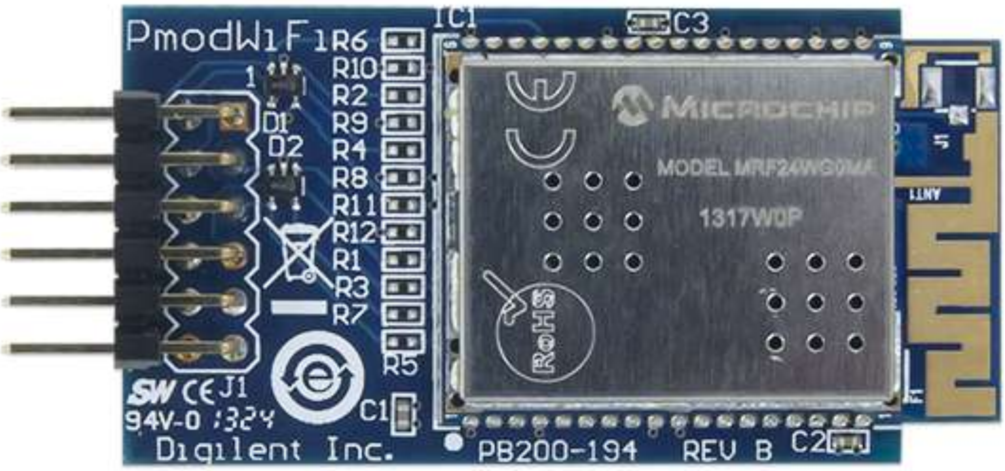


Figure 2. 12: Digital PMOD Wi-Fi Module

2.5.1 Features:

- IEEE 802.11-compliant RF transceiver
- Serialized unique MAC address
- Data rate: 1 to 11 Mbps for 802.11b / 6 to 54 Mbps for 802.11g
- Compatible with IEEE 802.11b/g/n networks
- Small size: 21 mm x 31 mm 36-pin Surface Mount module
- Integrated PCB antenna (MRF24WG0MA)
- External antenna option (MRF24WG0MB) with ultra-miniature coaxial (U.FL) connector
- Easy integration into final product – accelerates product development, provides quicker time to market

2.5.2 Operational:

- Single operating voltage: 2.8V to 3.6V (3.3V typical)
- Temperature range: -40°C to +85°C
- Simple, four-wire SPI interface with interrupt
- Low-current consumption:
  - RX mode – 156 mA (typical)
  - TX mode – 240 mA (+18 dBm typical)
  - PS mode – 4 mA (typical)
  - Hibernate mode – 0.1 mA (typical)

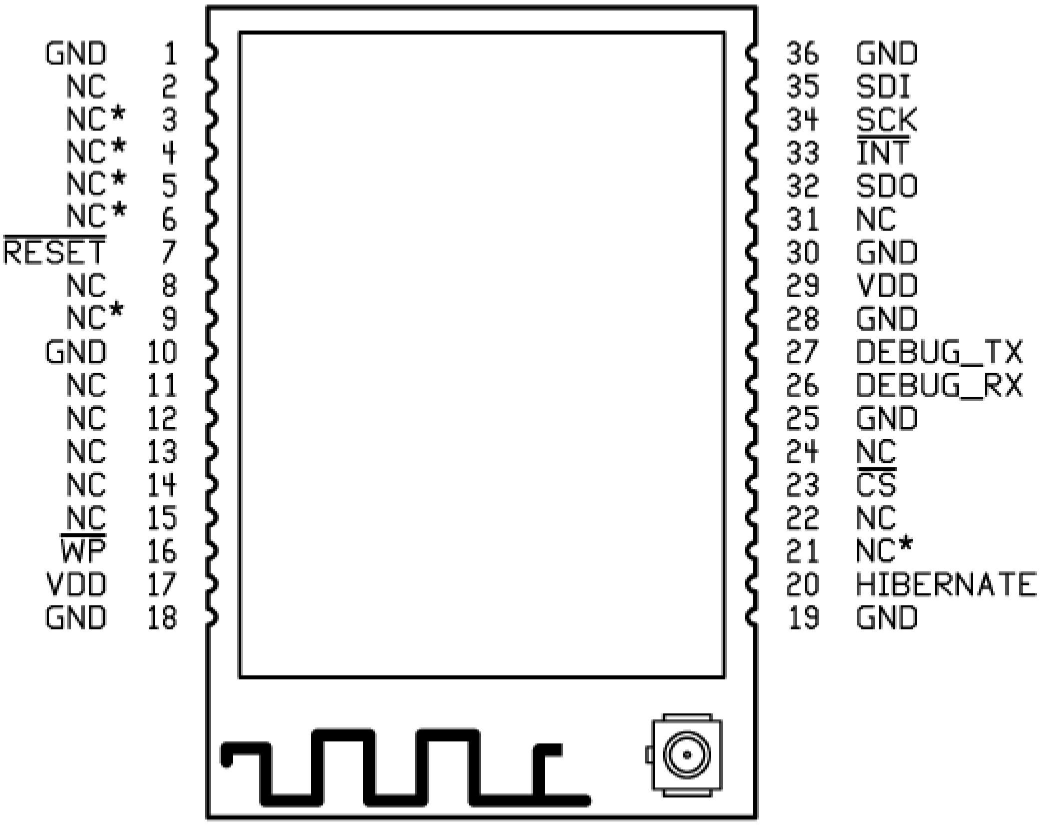


Figure 2. 13 Pin Diagram of Digital PMOD WiFi

Pin	Symbol	Type	Description
1	GND	P	Ground
2	NC*	NC*	No connect(3)
3	NC*	NC*	No connect(3)
4	NC*	NC*	No connect(4)
5	NC*	NC*	No connect(3)
6	NC*	NC*	No connect(3)
7	$\overline{\text{RESET}}$	I: Constant(1)	Module Reset input
8	NC	NC	Do not connect
9	NC*	NC*	No connect(3)
10	GND	P	Ground
11	NC	NC	Do not connect
12	NC	NC	Do not connect
13	NC	NC	Do not connect
14	NC	NC	Do not connect
15	NC	NC	Do not connect
16	$\overline{\text{WP}}$(2)	I	Write protect (this pin is used to enable FLASH update)
17	VDD	P	Power
18	GND	P	Ground
19	GND	P	Ground
20	HIBERNATE	I	Hibernate mode enable (high input will disable the module)
21	NC*	NC*	No connect(3)
22	NC	NC	Do not connect
23	$\overline{\text{CS}}$	I: Constant(1)	SPI Chip Select input, constant drive or pull-up required
24	NC	NC	Do not connect
25	GND	P	Ground
26	DEBUGRX	I	Serial debug port input (see **Section 2.0 "Circuit Description"**)
27	DEBUGTX	O	Serial debug port output (see **Section 2.0 "Circuit Description"**)
28	GND	P	Ground
29	VDD	P	Power
30	GND	P	Ground
31	NC	NC	Do not connect
32	SDO	O	SPI data out
33	$\overline{\text{INT}}$	O	Interrupt output (open drain – requires a pull-up)
34	SCK	I	SPI clock input
35	SDI	I	SPI data in
36	GND	P	Ground

Legend: Pin type abbreviation: P = Power Input, I = Input, O = Output, NC = Do Not Connect, NC* = No Connect

Figure 2. 14: Pin Description of Digital PMOD WiFi

Literature Survey

This chapter discusses the literature review from previous researches that are related to this project. The previous projects give some ideas and understanding on several previous projects. There are previous researches that have been done on detecting and monitoring the air quality either indoor or outdoor using various methods and discussed here.

3.1 Need of Dust Monitoring System:

Urban air quality is currently a global concern, which can be attributed to the massive scale of urbanization and population growth, together with their resultant increases in traffic, industrialization and energy use. It is understood that technological improvements in low emission motor engines have been offset by an exponential increase in vehicle numbers. Consequently, the release of pollutants into the atmosphere continues to increase, having adverse impacts on a local, regional and global scale, with significant associated health-effects. A recent 'Global Burden of Disease' study has provided new evidence of the significant role that air pollution plays globally, placing it among the top ten risks faced by human beings. Many of the world's cities are unable to comply with the prescribed concentration limits of air pollutants, and in many cases, reported measurements far exceed them, resulting in millions of premature deaths. At the forefront of pollutants which exceed concentration limits are coarse (PM10) and fine particulate matter (PM2.5), and unregulated ultrafine particles (b100 nm), making this issue even more complex. For example, a recent World Health Organization report on ambient air pollution suggests that the annual mean concentration of PM10 has increased by more than 5% between 2008 and 2013 in 720 cities across the world. A reduction in long-term exposure to PM10 by 5µg per cubic meter in Europe has been reported to "prevent" between 3000 and 8000 early deaths annually. Similar estimates for PM2.5 suggest an average loss of 7–8 months in life expectancy for UK residents and about £20 billion per year in corresponding health costs. An equivalent estimate for exposure to ultrafine particles, which have a greater potential for adverse health impacts compared to their larger counterparts, is currently unavailable, but will further increase the health and economic burden in the UK and elsewhere.

Air quality varies over a relatively small scale since the resulting pollutant concentration in a specific place depends predominantly on local emission sources and atmospheric flow conditions. The flow of air masses in urban environments is typically turbulent and difficult to predict without sophisticated numerical modeling tools. Real-time high resolution (b1 m) pollutant concentration maps for large urban areas do not exist at present because they require a large amount of data, computing facilities and input details that are not available for many cities. This complexity makes the assessment of actual human exposure to pollutants challenging. One solution to overcome the lack of small-scale, high-precision measurements of air quality is to adopt low-cost methods for robust environmental surveillance. Although these methods tend to produce lower quality data, they are able to be used in a high number of locations simultaneously, which allows for high-resolution exposure assessment mapping of city environments.

3.2 Measurement Principles for Dust Detection:

The common measurement principles for dust detection including Gravimetric measurement, Triboelectric measurement and Optical measurement. The gravimetric principle is the method used in analytical chemistry for quantitative determination of analysis based on the mass of a solid. [5] It is suitable to measure concentration of dust in liquids. The Triboelectric measurement is another measurement of dust concentration that contact electrification of certain material become electrically charged after they come into contact with another different material and are then separated. This principle can be applied in the measurement of dust for ceramic, bags, and cartridge filters or cyclones where indicative monitoring is required. The optical measurement of dust based on the attenuation of the intensity of a light beam by absorption and dispersion penetrating a cloud with solid particles.

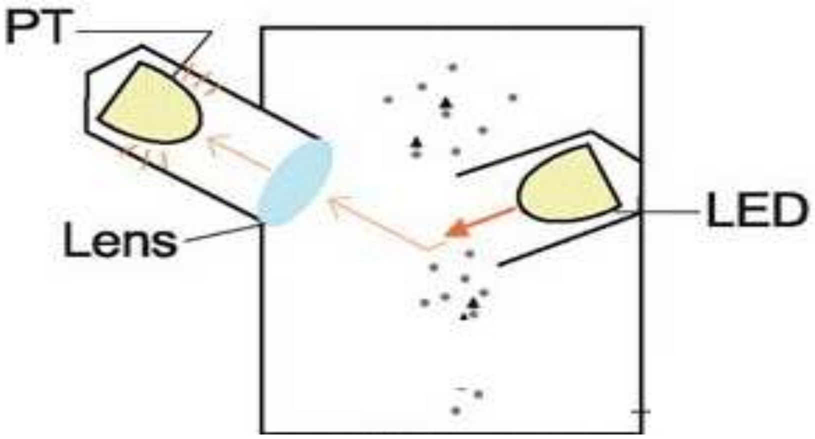


Figure 3. 1: Measurement Principles for Dust Sensor

3.3 Related studies:

Srinivas D., [11] developed Public Transportation Infrastructure such as buses, which have fixed and reliable routes along high volume corridors using a custom-made Mobile Sensing Box (MSB). MSB includes a microcontroller board with add-on sensors, a peripheral GPS receiver and a cellular modem. Connecting to the bus battery would provide the power supply needed to operate this model. Used two pollution sensors to measure carbon monoxide and particulate matter concentrations.

Adrew L., [12] designed and implemented a new generation of detectors, nanotechnology based metal oxide semiconductors such as ZnO semiconductor to substitute the typical analytical tools and adapt or extend the air quality monitoring system. In fact these solid state gas sensors offer an excellent opportunity for implementation in environmental monitoring due to light weight, extremely small size, robustness, low cost and also as they can be installed anywhere to collect data covering extensive areas. The air quality data can eventually be transmitted through a Wireless GIS network system to the general public.

Kim, S. and Paulos, E. [13] designed and implemented a system called Air that able to measure, visualize, and share indoor air quality. A DC 1100 air quality is used to measure the level of indoor pollutant, an AVR-based Arduino which transplant inside the air quality monitor, and an iPod Touch is used to process, visualize and transmit the data wireless to the Arduino. The data will be reported every 15 seconds at the same time the Arduino will encode the data into a series of audio tones like a modem and will be read by iPod Touch via microphone port. The data can be shared from central server in real-time by using Wi-Fi networking.

Rajiv P., and Khedo K., [14] presented a wireless sensor network air pollution monitoring system (WAPMS). The sensor networks are dense wireless networks of small, low-cost sensors, which collect and disseminate environmental data. Wireless sensor networks facilitate monitoring and controlling of physical environments from remote locations with better accuracy. They have applications in a variety of fields such as environmental monitoring, indoor climate control, surveillance, structural monitoring, medical diagnostics, disaster management, emergency response, ambient air monitoring and gathering sensing information in hospitable locations. Its system design more easy to user with develop an architecture to define nodes and their interaction, Visualization of collected data from the WSN using statistical and user-friendly methods such as tables and line graphs, generation of reports on a daily or monthly basis as well as real-time notifications during serious states of air pollution for use by appropriate authorities and Provision of an index to categorize the various levels of air pollution, with associated colours to meaningfully represent the seriousness of air pollution.

Zakaria, N., S., [15] designed a low cost API system for a Rice Mill Factory located in Pering, Kedah Darul Aman. The factory needs to provide a good air quality surrounding not only for its employee but also for the neighbouring villagers. A 24 hours production time also contributed to its pollution factor and the level of exposure endured by the employee during the work shift needs to be considered as well. Rice Husk/Dust Air Particle Sensor using ZigBee Wireless Sensor Network was developed using SHARP GP2Y1010AU0F optical dust sensor as the measurement tools, an Arduino Fio board was used as the development board for its expansion capabilities using ZigBee Wireless Modules. A point to point approached was developed with the data being transmitted back to its host computer, and a serial port was used to read the HEX string data. Parallax Microcontroller Acquisition to Excel (PLX DAQ) software is used to read the string data and save it to Microsoft Excel software. Using Visual Basic Application on Microsoft Excel, a graph displaying the dust measurement can be viewed on the real time basis.

Conclusively, the sensor and methods used for this project was substantial enough to monitor the dust density for reducing the dust pollution in the Rice Mill Factory.

Sung, J.O and Wan, Y.C [16] developed a RF wireless sensor module with optimal communication condition to monitor indoor air quality in a room or office. The monitoring work can be done by web-based monitoring system together with other home networking system by using PDA (Personal Digital Assistant). There are several sensors for instance; temperature sensor, humidity sensor, Carbon Dioxide sensor and flying dust sensor were built in the RF transmitter board for monitoring the room environment. An Intel 8051 microcomputer was used to control the power switches of consumer electronics through signals received from PC or PDA.

Phang Qili [17] has presented a room dust monitoring system. The main objective of the system was to monitor the dust concentration of a room and show the readings on a personal computer in real time. These systems used an Arduino Uno controller work based on a Shinyei PPD20V particle sensor to measure the dust concentration in a room. The readings taken from the sensor will be sending to computer to show in real time using Graphical User Interfacing (GUI) by using Visual Basic program. The result was taken in several conditions including clean room, dusty room, room with cooking haze and room with cigarette smoke. The result varied with the condition of the room. With this device, the awareness of the effects of human activities on indoor pollutants can be rose up and thus lead to human health and well-being.

Problem Statement

This section discusses about the problem definition and problem statement after finding the gaps in the literature review. The objectives of the project obtained from the problem definition have been described. The methods and methodologies for implementing the objectives have been described.

4.1 Introduction:

As per the literature survey, dust monitoring system is important required system in smart cities. Nowadays people are paying more attention towards their personal life as people are facing with breathing problem involving asthma problems. This system it gives importance in monitoring the dust in an area and hence providing the alertness to the people. Recording and precise measurement of dust and particle emissions are required for sustainable environmental protection. This is because dust particles, particularly those created by industrial plants, vehicle pollution have significant impacts on humans and the environment. Nowadays, individual who wants to have an API system has to pay a fortune because the installation cost is very high and expensive. The size is also large and not portable. These systems are usually used by government's bodies and large companies who own industrial plants to monitor the air quality. This project has been proposed to produce a cheaper version of API system and portable using cheaper materials costs, readily available dust sensor, FPGA board and Arduino microcontroller.

4.2 Aim of Project:

To design, develop and implement an IoT based Dust density monitoring system using FPGA.

4.3 Design Objectives:

- To perform literature survey on existing dust monitoring system, dust sensors, Wi-Fi interface and memory interface with FPGA, Android Application interface, webpage development
- To develop and configure interface unit for Wi-Fi module, dust sensor, ADC, SD card
- To develop and configure Android application for the proposed dust monitoring system
- To develop and configure webpage with http server
- To configure http server with FPGA, integrate and verify the proposed dust monitoring system

4.4 Methods and Methodologies:

- Literature survey on dust sensors, Wi-Fi interface and memory interface with FPGA, Android Application interface, webpage development
- Specifications of the selected sensor, Wi-Fi module has been identified
- Functional verification and timing analysis of behavioral model of dust sensor has been performed

- Configuration and implementation of ADC, Wi-Fi module, webpage with server
- The MicroBlaze design based IP integration and configuration of various peripherals has been performed
- The Xilinx SDK 2016.3 has been used for configuration and implementation of ADC, Wi-Fi module, webpage with server
- The Eclipse Android Development Kit has been used for android application development
- SD Card and Android application configuration for the proposed system

Design and Development of Behavioral Model Dust Sensor

5.1 Development of Semi- Automated Calibration System:

This section discusses about the design of proposed dust monitoring system and Functional verification and timing analysis of behavioral model of dust sensor.

5.1.1 Design of Proposed System:

Now a days we could see so much of pollution around us especially the air pollution which is leading to the more and more hazardous diseases to the human health. In our daily routine we would have faced so much dust in the environment resulting in dust allergies, cough and inflammation in nose, throat infections and respiratory diseases. So there is a need for changing this dusty environment to clean environment. To avoid dust around basically those dust particles has to be monitored which gives an idea about the concentration of dust. In this project, an effective solution for monitoring the real time dust values using the wireless sensor networks.

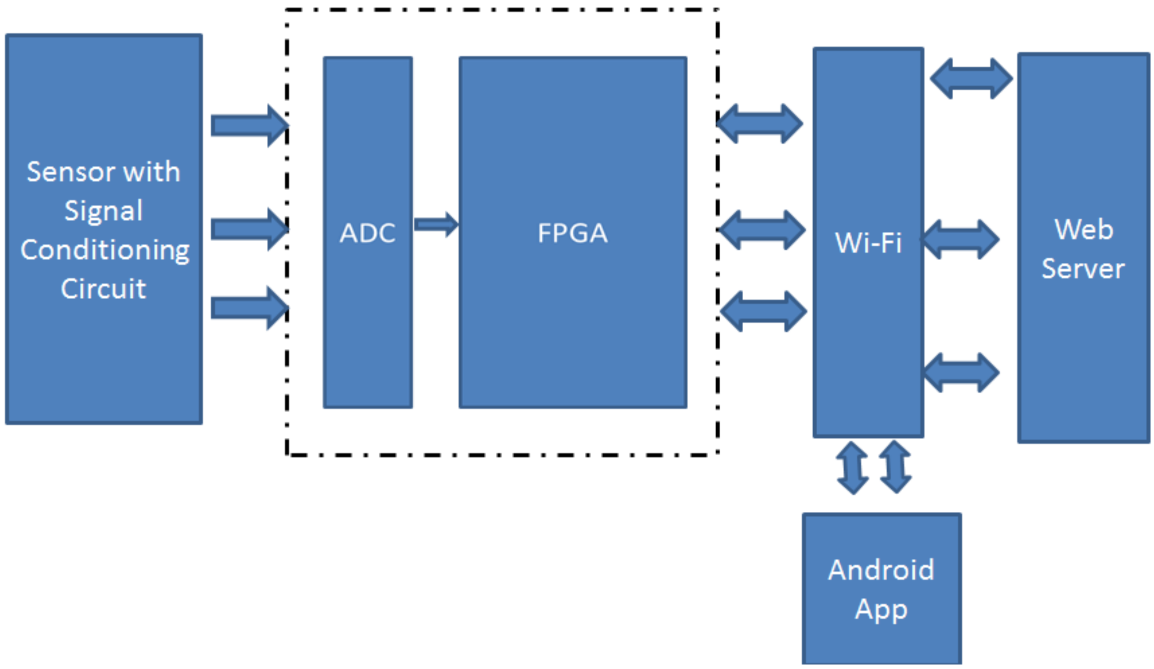


Figure 5. 1: Design of Proposed System

The Figure 5.1 shows the block diagram of the proposed system, the system uses a dust sensing entity called an optical dust sensor. The sensed values of the sensor are in analog format, these values are converted in to digital format by the analog to digital converter, which is in built in the Nexys 4 FPGA board.

FPGA monitors and controls the display of sensed values of dust sensor on to the wireless android device using the recent Wi-Fi technology.

5.1.2 Functional Verification and Timing Analysis of Behavioral Model of Dust Sensor:

In this section, the functionality of model of dust sensor is verified and timing analysis of output waveform is done. The circuit which provides the behavioral model of dust sensor is built and simulation is for the same. This analysis is done using NI-MultiSim 13.0.

5.1.2.1 Behavioral Model of Dust Sensor:

The circuit in Figure 5.2 is a simple behavioral model of GP2Y1010AU0F dust sensor. The model has a 7th pin which is not present in the real device. This pin, DUSTIN, is a high impedance input used to allow a voltage representing dust density in mg/m^3 to be supplied to the model in order to generate an output voltage in response to a pulse input to the LED driver circuit. The DUSTIN input is scaled so that 0.1V represents 1mg/m^3.

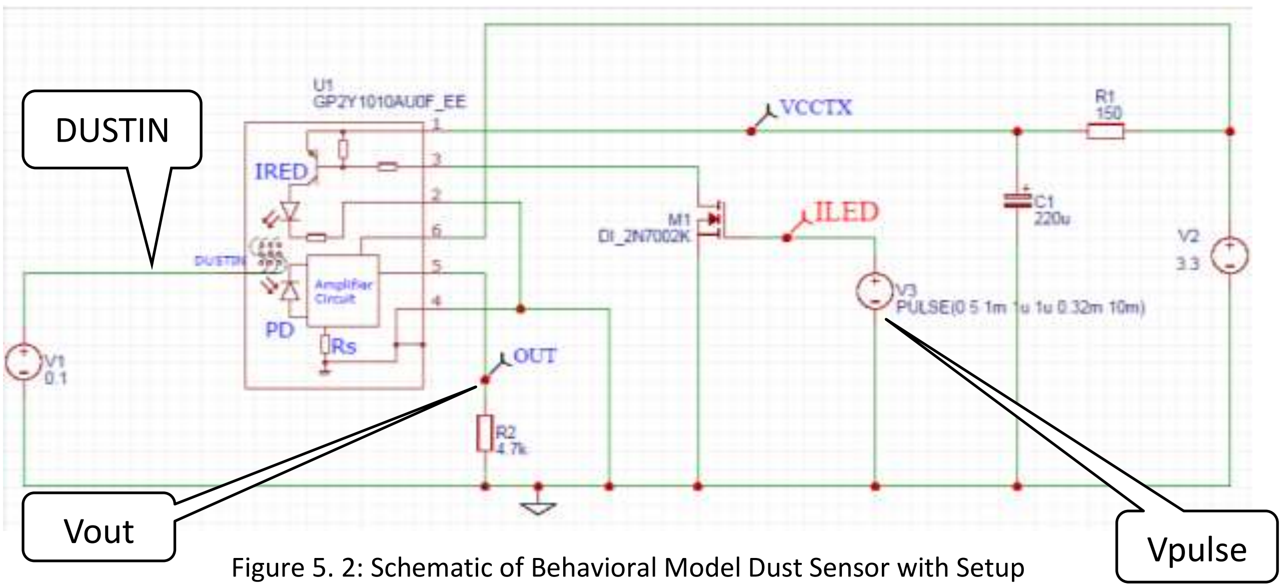


Figure 5. 2: Schematic of Behavioral Model Dust Sensor with Setup

The output is scaled at 5V/(mg/m^3) with an offset, VOC of 0.9V and a maximum saturated output VOH of 3.6V. Output resistance is not specified in the device literature but is set in this model to an arbitrary 100R. In the opinion of the model designer, the current drain of the LED driver stage is very poorly specified and the datasheet definition of the LED terminal current of ILED is misleading. The LED control input pin (pin 3) is supposed to be a 'low power' control input. Given the circuit shown for the LED driver, a control current of 10mA (Digilent, 2013) would imply that the LED current being controlled is much larger: in the region of 100mA or maybe more. The external circuit of 150R and 220uF together with the quoted

0.32ms pulse width at 10ms period. It is therefore likely that the current drawn by the VLED pin (pin 1) is something like 110mA (100mA for the IRED itself plus 10mA for the current out of the LED pin) for the duration of the LED being on.

Hence for the purposes of this model, an arbitrary LED current of 100mA has been assumed together with the quoted ILED of 10mA. Both of these currents are assumed to be zero when the LED is off.

5.1.2.2 Simulation of Model of Dust Sensor:

The simulation results of behavioural model of dust sensor are shown in Figure 5.3. The model is taking 0.2ms to reach its maximum value dust particle. The sensor also producing the no dust voltage(Vnd) which is important factor for processing sensor data in FPGA and for calculating the dust density.

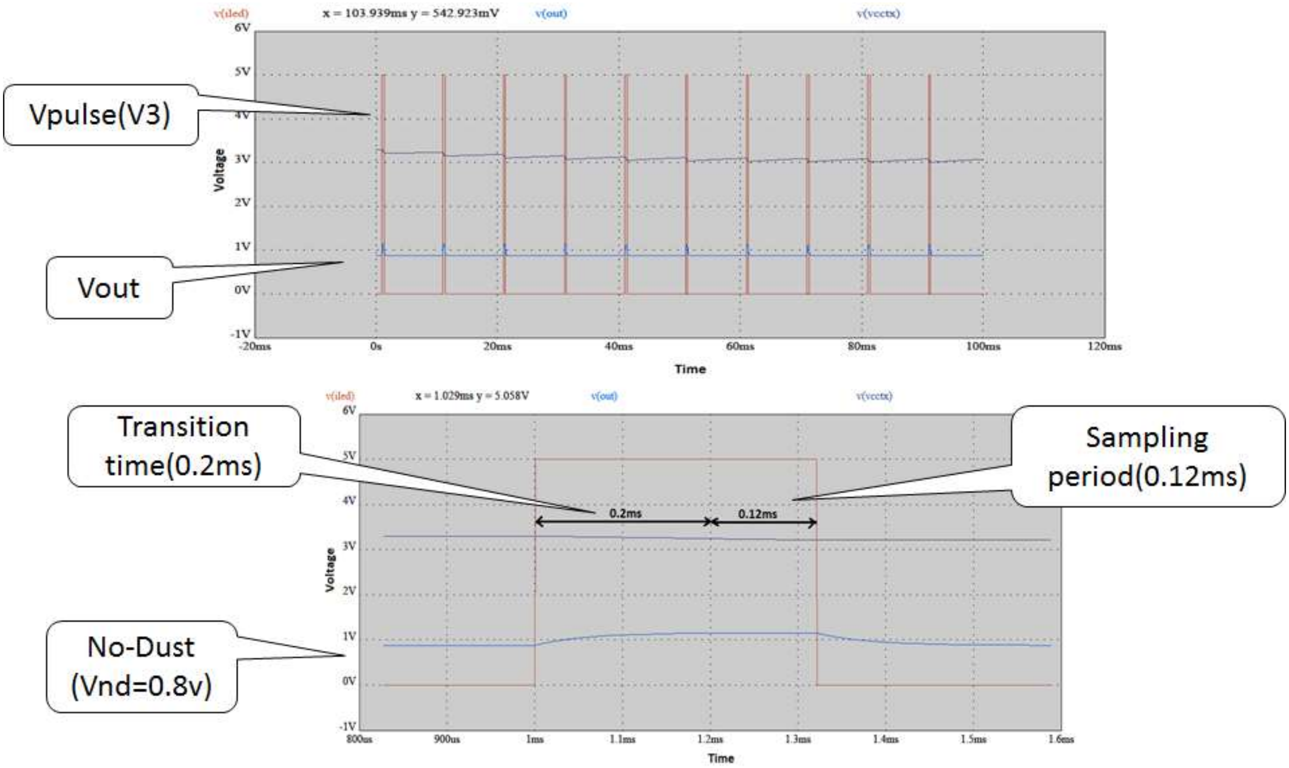


Figure 5. 3: Simulation of Behavioural circuit

From all these observation of simulation results, the following conclusions are drawn.

- The Processor should give 0.2-0.3ms delay after that it should start to take sample voltage from pin-5 of sensor(Vo)
- There is period of 0.12ms for sampling. When finished, set the pin-3(ILED) to low to disable the internal infrared emitting diode

For every dust particles present on the sensor, the sensor generates a voltage, with this voltage the dust density can be measured in mg/m^3.

Table 5. 1: Dust Density Values

Practical Values		Simulation Values	
Voltage (V)	**Dust density(mg/m^3)**	**Voltage (V)**	**Dust density(mg/m^3)**
0.90	0.053	0.8	0.040
0.95	0.061	0.9	0.053
1.00	0.070	1.0	0.070
1.25	0.110	1.1	0.087
1.50	0.120	1.2	0.104

The Table 5.1 tells about the variable voltage values generated corresponding to the dust particles blown on the optical dust sensor. The dust density values shows the concentration of dust present in the particular given area.

Pmod XADC Configuration

6.1 Introduction:

It is a 12 pin Pmod connector provides two 3.3V VCC signals (pins 6 and 12),two Ground signals (pins 5 and 11) and eight logic signals. VCC and Ground pins provides a current of 1A. Pmod data signals are not automatically routed, they are routed using available tracks and techniques. Sensors, Motor drivers, A/D’s, D/A’s can be added to the Pmod expansion connectors by using the collection of Pmod accessory boards. Figure 6.1 shows the pin diagram of the Pmod JXADC port of the Nexys 4 board.

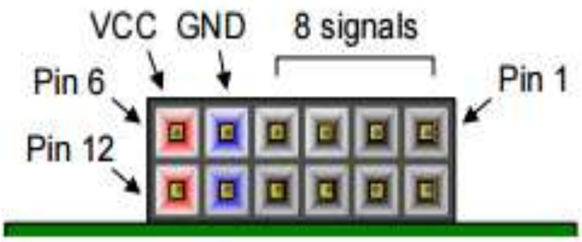


Figure 6. 1: Pin diagram of JXADC port

The Pmod ACL IP is a dual channel 12-bit analog-to-digital converter capable of operating at 1 MSPS. Either channel can be driven by any of the auxiliary analog input pairs connected to the JXADC header. The XADC core is controlled and accessed from a user design via the Dynamic Reconfiguration Port (DRP). For communication over Wi-Fi the digital data coming from the PmodACL is stored in registers and in SDK platform it should be read and transmit over Wi-Fi. Its result appears on web based sever and android application.

6.1.1 ADC- Analog to Digital Converter:

ADC is used to convert from analog signal to its equivalent digital form. The input to ADC will be in analog voltage form and the output will be obtained in binary form. In nexys-4 board it consists of a built-in system monitor function. The user can access up to 12bit, 1-MSPS ADC. It has on-chip sensors and on-chip power supply measurement. (Frost, R., 2013) Input voltage range to ADC is from 0-3.3V range. It can support up to 17 analog input signals. ADC basically measures voltage drop between analog pair pins V_P and V_N which provides the respective digital values.

6.1.2 ADC Specifications:

- Can access up to 12bit, 1-MSPS ADC
- Analog input range to ADC is from 0-3.3V range
- General purpose analog inputs

- 17 external analog inputs channel can be selected
- On-Chip temperature and power supply measurement
- Full access from fabric or JTAG TAP to System Monitor
- Fully operational prior to FPGA configuration and during device power down (access via JTAG TAP only)

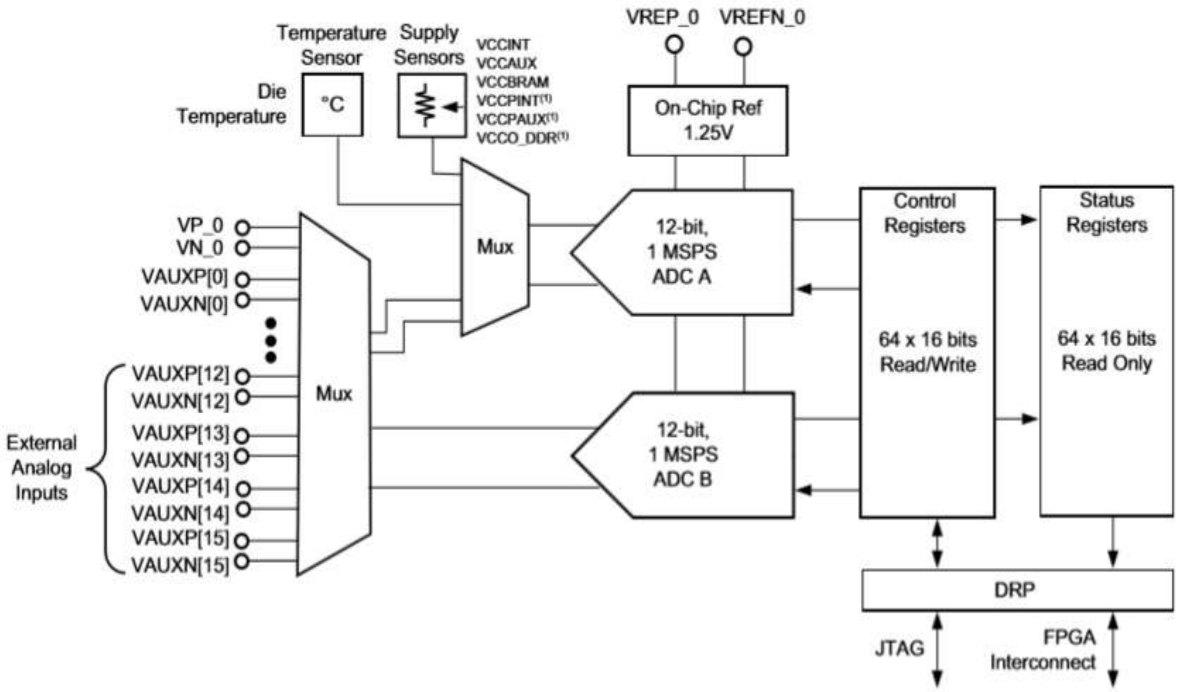


Figure 6. 2: Block diagram of Nexys-4 system monitor

Figure 6.2 shows the block diagram of nexys-4 System Monitor which includes temperature sensor, ADC, supply sensor, Register File interface. The system monitor consists of 12 bit ADC which also consists of temperature sensor. Analog input pair pins are V_P, V_N which are accessed to external voltages and user selectable auxiliary pins (16 bits) are also supported. External analog pins are provided to measure the physical environment. System monitor is fully functional when the system is power up and data measurement is taken from JTAG port. (Sources, A.E., 2012) sThe measured values are obtained after the End of conversion (EOC) OR End of sequence, which becomes high after ADC conversion.

6.1.3 ADC Transfer Functions:

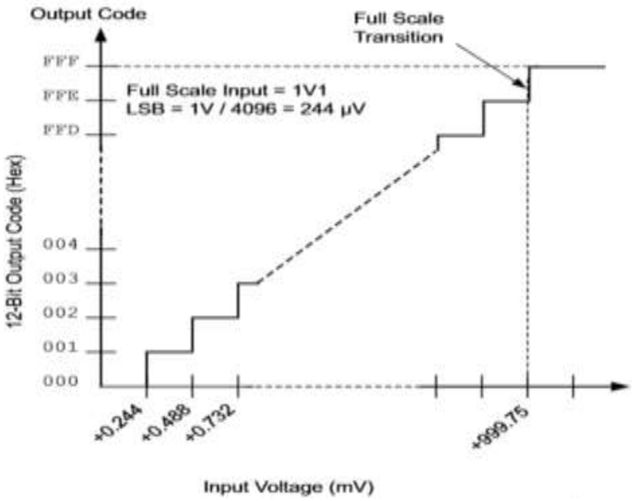


Figure 6.3 Transfer function of ADC

ADC provides the digital output of the given analog input signal. The ADC produces a full-scale 12-bit code (FFFh) with a 1V differential input voltage on its external analog inputs as shown in Figure 6.3.

FPGA Implementation of Interface Control Unit

7.1 Introduction:

The proposed architecture is implemented on FPGA for that MicroBlaze soft processor based embedded system design of a Web server, which is designed using the Embedded Development Kit (EDK) procedures of Xilinx Vivado Design Suite 2016.3.

In this section the embedded platform is developed and implemented to set up a system as a Web client to connect to the Web server running on the MicroBlaze processor.

7.1.1 Architecture of Interface Control Unit Block:

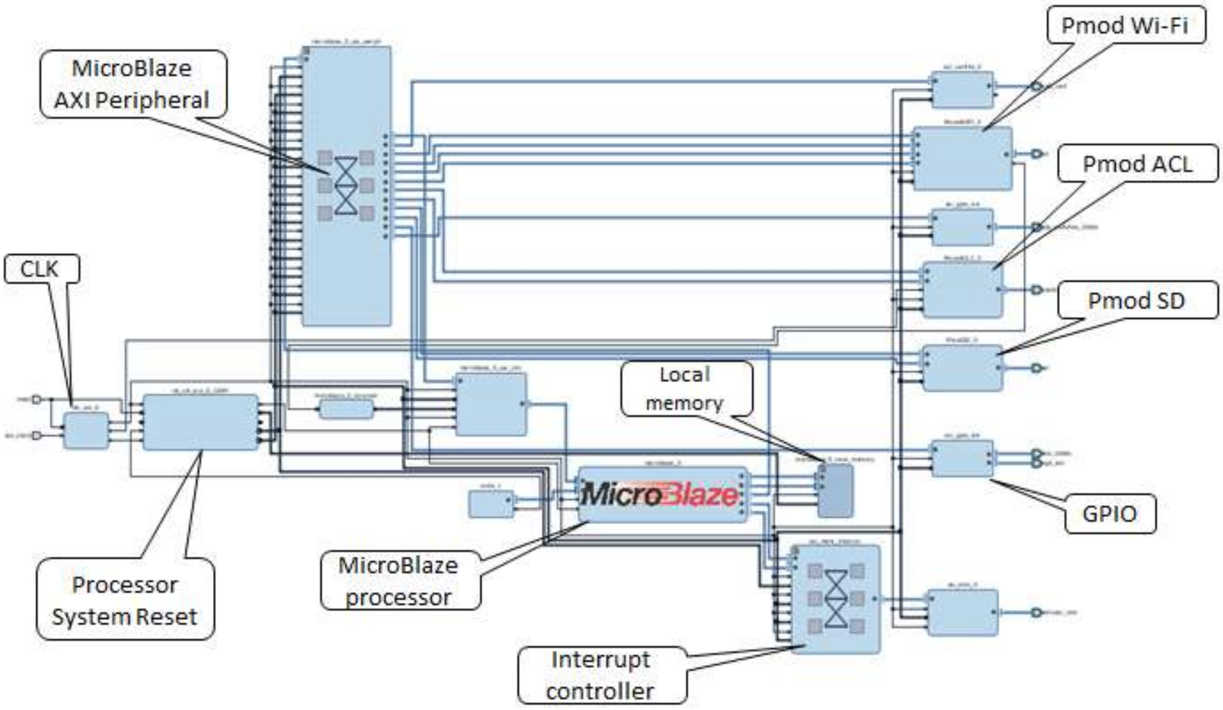


Figure 7. 1: Architecture of Interface Control Unit

The architecture of Interface control unit in Figure 7.1 consist of MicroBlaze Processor, MicroBlaze AXI Peripherals, Interrupt Controller, Pmod ACL, Pmod Wi-Fi, Pmod SD, General purpose input output pins. All the block mentioned above are configured with the MicroBlaze processor by using the IP blocks.

MicroBlaze Processor:

MicroBlaze is a soft IP core from Xilinx that will implement a microprocessor entirely within the Xilinx FPGA general purpose memory and logic fabric. As a soft-core processor, MicroBlaze is entirely implemented in the general-purpose memory and fabric of Xilinx FPGAs.

MicroBlaze AXI Peripherals:

MicroBlaze AXI Peripherals supports data transfers between the Interface and external synchronous and peripheral devices (World Health Organization, 2013) such as USB and LAN devices, which have processor interface inside them providing communication between the devices.

Interrupt Controller:

Interrupt Controller is a device which combines many sources of interrupt on one or more CPU lines, allowing priority levels so that they can be assigned to its interrupt outputs

Pmod ACL:

Pmod ACL is a 3-axis digital accelerometer, the module provides input to the system board. The Communication is possible through SPI or I2C using the on-board Analog Devices ADXL345 which allows to access up to 13 bits of data.

Pmod Wi-Fi:

Pmod Wi-Fi is an interface board developed by the Microchip Wi-Fi module. It is designed to use with the Microchip microcontroller and the Microchip TCP/IP Stack. The primary communications with the Pmod Wi-Fi is an SPI bus.

Pmod SD:

Pmod SD provides a full sized SD card slot that can be accessed through SPI. Using which provides Store and access large amounts of date from system. This allows 1-bit and 4-bit communication from the system for accessing the data from the card for configuring the web page.

GPIO:

General purpose input output is an interface which provides an easy access for the devices. Generally there are multiple GPIO pins for the use of multiple interactions for many simultaneous applications.

General Design Flow for Vivado:

Step 1: Open Vivado and select Nexys 4 board

Step 2: Create a new Vivado Project

Step 3: Create empty block design workspace inside the new project

Step 4: Add required IP blocks using the IP integrator tool and build Hardware Design

Step 5: Validate and save block design

Step 6: Create HDL system wrapper

Step 7: Run design Synthesis and Implementation

Step 8: Generate Bit File

Step 9: Export Hardware Design including the generated bit stream file to SDK tool

Step 10: Launch SDK

The Figure 7.2 showing the Instruction and data cache of 32kb each, usually the data in cache cannot be accessed. The area, performance of the device is also displayed indicating the frequency of 80MHz.

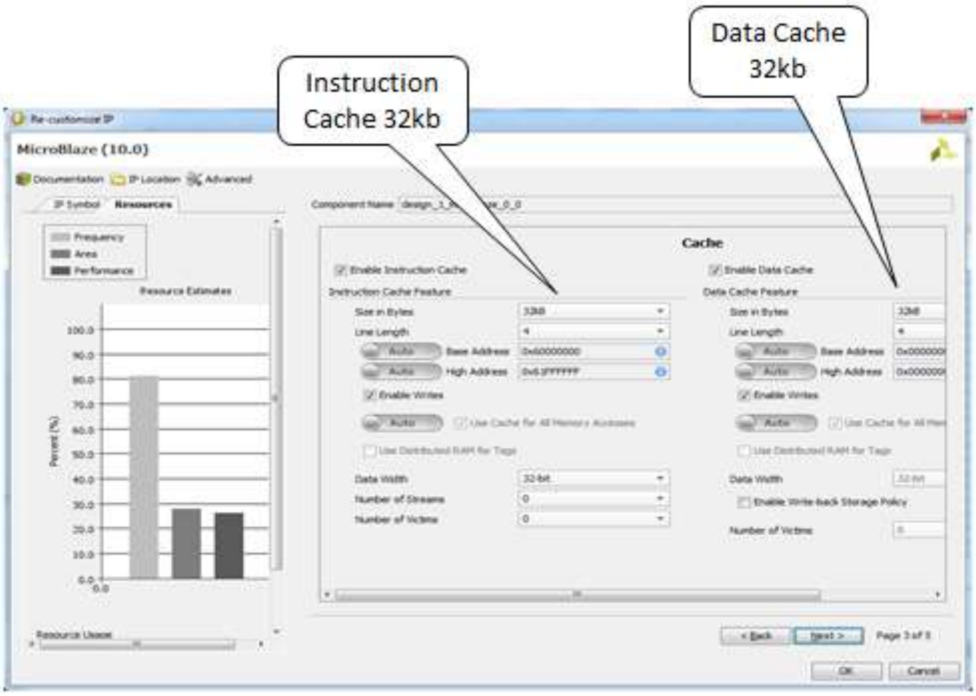


Figure 7. 2: IP Configuration

Clock Wizard Block Setup:

Basically the designed system uses two clocks one is for entire system design and another is for Pmod Wi-Fi device. The Figure 7.3 shows the clock distribution in which the system takes 100MhHz and the Pmod Wi-Fi device takes 25MHz.

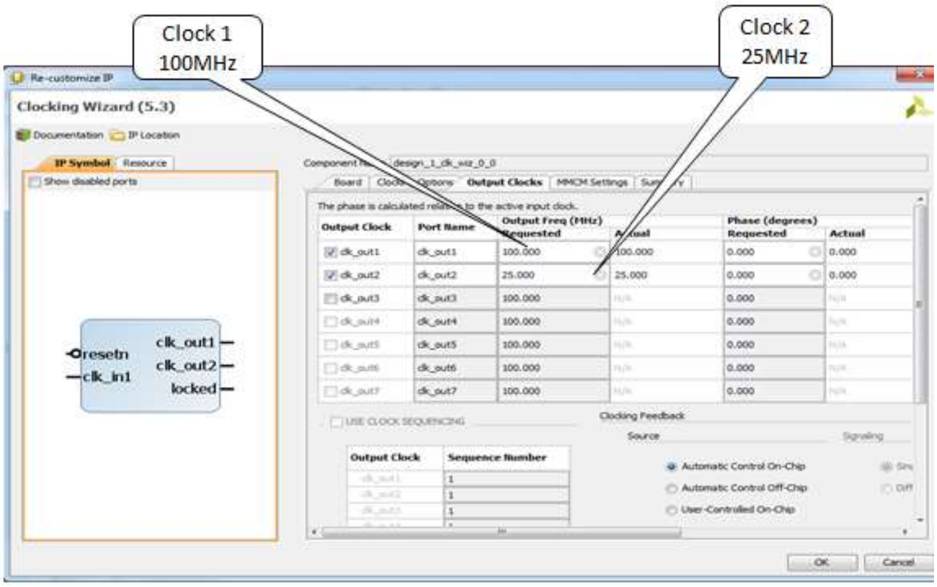


Figure 7. 3: Clock Configuration

Concat Block Configuration:

The number of port for concat block is configured as 1 which is further connected with the Pmod Wifi block WF interrupt in port for processor interrupt handling purpose. Figure 7.4 shows the concat block configuration with the MicroBlaze design.

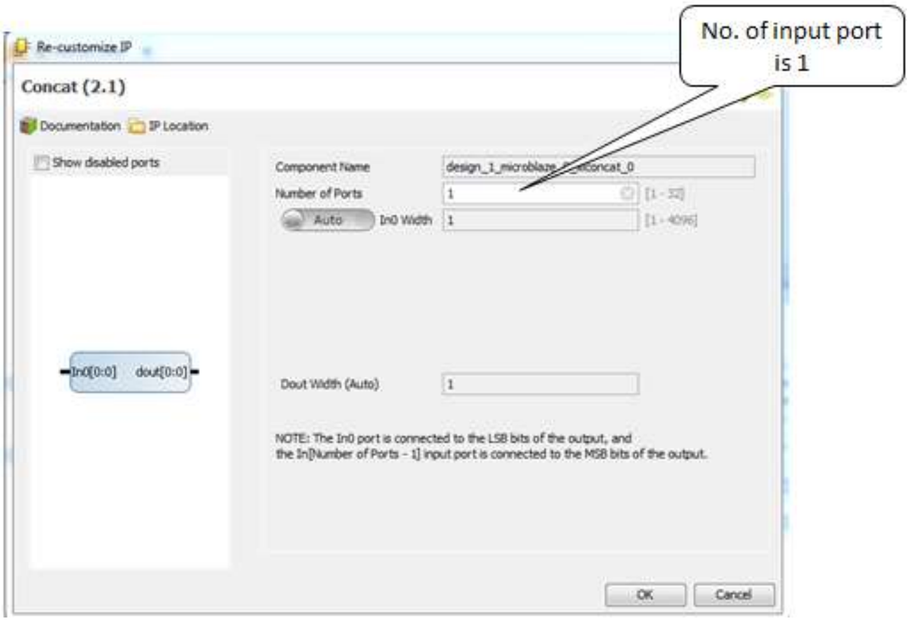


Figure 7. 4: Concat Block Configuration

AXI UART Configuration:

The Baud Rate is the rate at which the information is transmitted. The data is sent for UART at 115200 baud rate and data bits is set to 8 bits. The operating frequency of the device is set to 100MHz. The

Serial communication of data is carried out using SPI protocol and AXI protocol. Figure 7.5 shows the Baud rate setup for the data transmission.

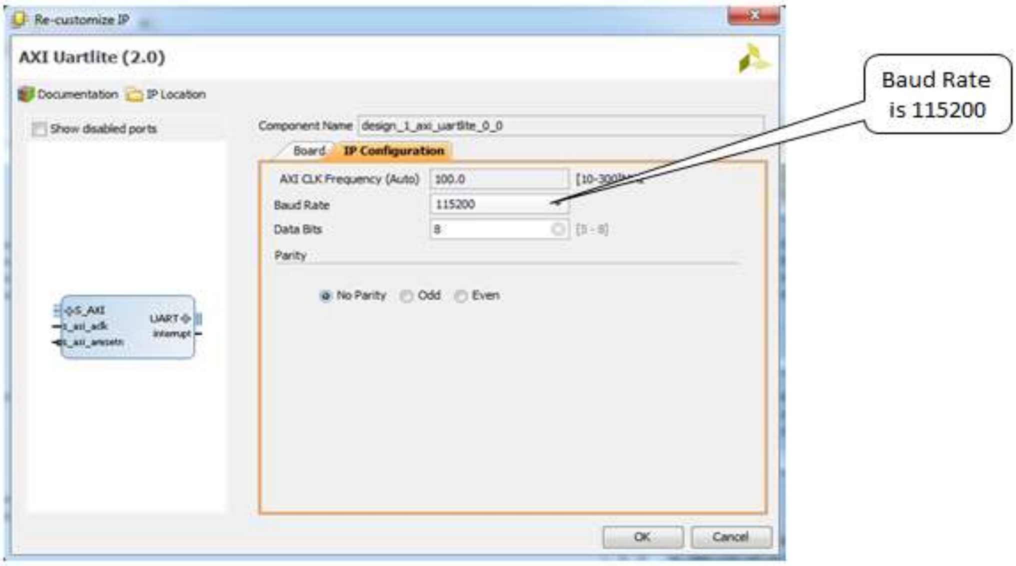


Figure 7. 5: AXI UART Configuration

After the configuration of all the blocks with the MicroBlaze processor, a system will be developed by interconnecting all the blocks of the design. Figure 7.6 shows the Routing and placement of blocks like MicroBlaze AXI Peripherals, Interrupt Controller, Pmod ACL, Pmod Wi-Fi, Pmod SD, General purpose input output pins with the MicroBlaze processor.

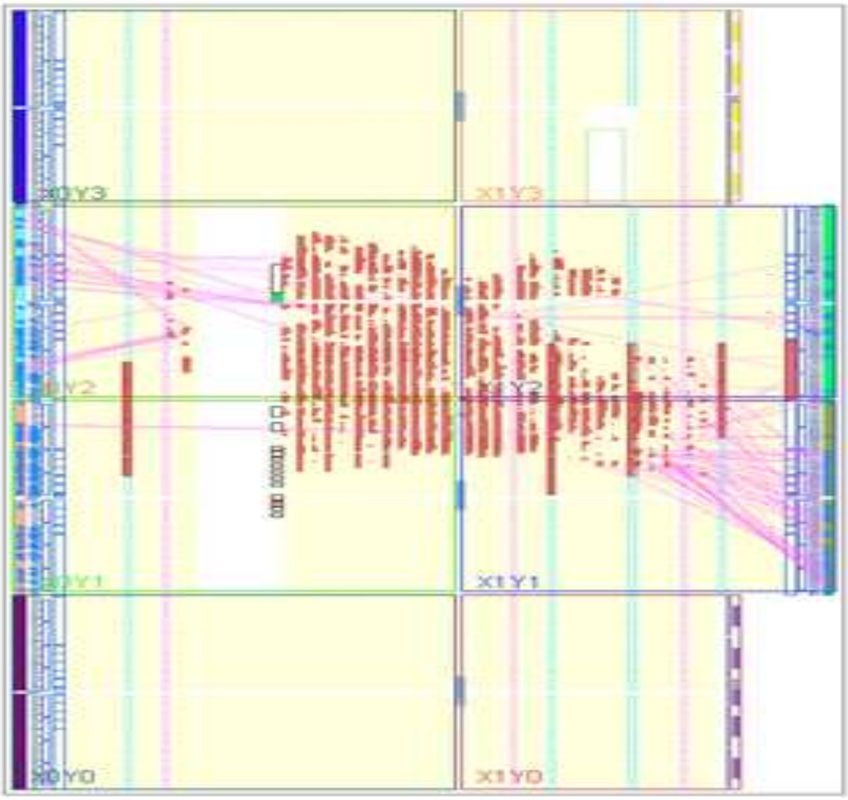


Figure 7. 6: Routing and Placement of different blocks with MicroBlaze

Implemented Design Power Summary:

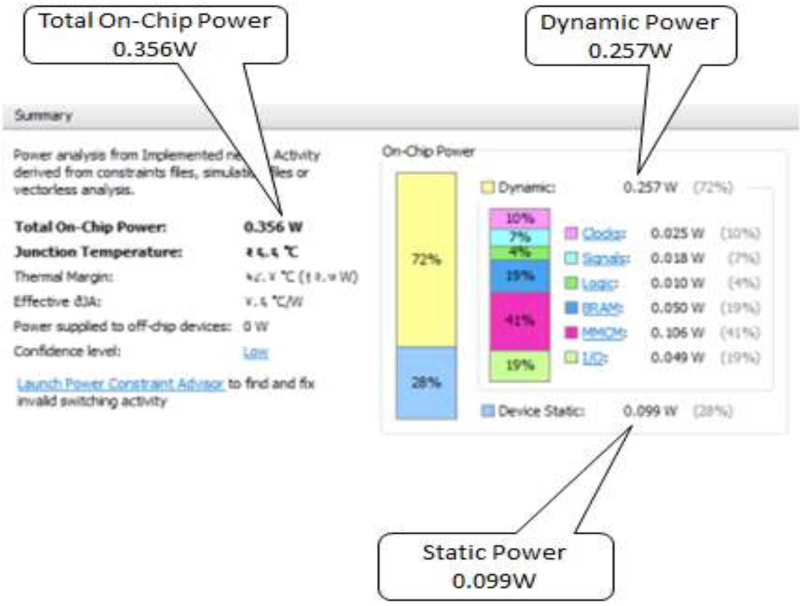


Figure 7. 7: Power Summary of the design

The total power is the sum of the dynamic and static power. Always the dynamic power will be more. Same is the case here, the dynamic power consumed by the device is 0.257 watt and static power is 0.099 watt. Figure 7.7 shows the power consumed from the design to carry out the desired operation. Lesser the power consumed more will be the advantage.

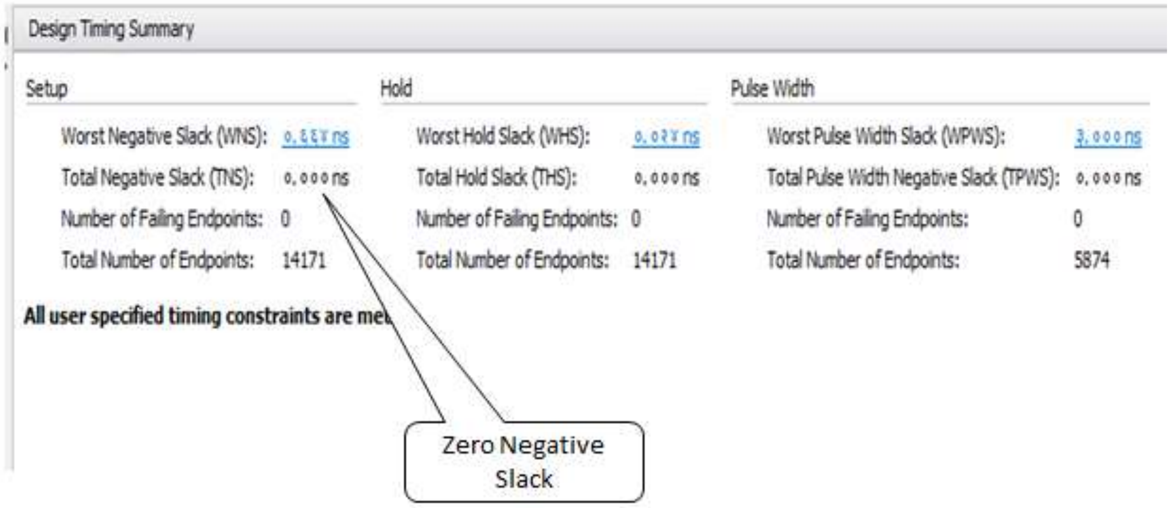


Figure 7. 8: Timing Summary of the design

For every design the timing matters the most. The delay has to be reduced as far as possible to increase the performance of the design. Timing path slack determines if the design is working at the specified speed or frequency. Slack should be positive or zero, Negative slack indicates the timing violation. Figure 7.8 shows the Timing summary of the design which has a zero negative slack.

CHAPTER 8

HTTP Server Configuration

8.1 Introduction:

When a web user uses the web browser and initiates the communication by sending the request for a specific set of information by making use of web server. If the resource is available the web server will respond with the content of requested resource or else the server sends back an error message. Web servers are used to serve World Wide Web. Web server is a computer system which stores, process and delivers the web pages to the clients. The communication between the client and the server follows a protocol called Hyper Text Transfer Protocol (HTTP) and it is shown in the below Figure 8.1.

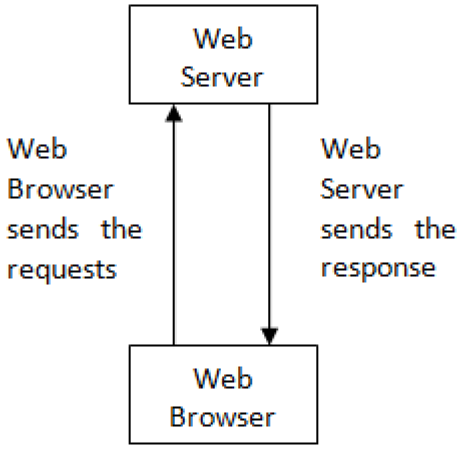
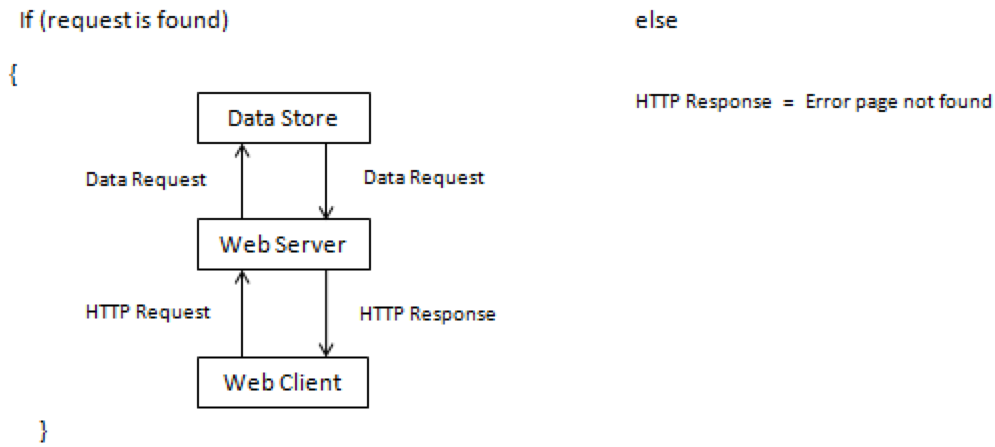


Figure 8. 1: HTTP Server

Hyper Text Transfer Protocol is an application protocol, many web servers uses this protocol to transfer the html pages. HTTP functions as a request - response protocol in client - server computing model. HTTP is text based streaming protocol that uses TCP (Norhafizah, R., 2015) connections for network transport. When the client submits the HTTP request to server, the server provides resources and returns the HTTP response to



the client. The response may contain the completion status or the requested content in the message body. If the requested content is not available then the server throws an error message in the message body.

HTTP is a connectionless, media independent, stateless protocol. Once the request is initialized and made from the browser, the connection between the client and server is removed and the client waits for the response from the server, the server processes and reestablishes the connection with the client and sends the response.

HTTP requests and responses consists of three main parts like Request line, Header lines, Message body. Request line specifies what kind of request or response message. The number of Header lines are variable if required. Request line and headers are text lines within <CF><LF>. Header line and the message body is separated by a blank line. Message body is optional, because few web pages will not present the message body. Message body can contain the status of reception of response from the server, it can also contain text or binary data. Type of data, length of data, language of the data all these details are present in the header lines descriptions. The Request line consists of Request Method, Request - URI, protocol version, ending with CRLF.

Request - line =Request Method SP Request - URI SP Protocol version CRLF.

HTTP Request Methods:

- GET: Request the URL resource
- POST: Sends data to the server
- HEAD: Same as GET, but transfers only status line and the header
- DELETE: Removes all URL resources
- TRACE: Performs message loop back test and path to resource
- OPTIONS: Communication for target resource
- CONNECT: Establishes a connection to the identified resource
- PATCH: Applies modifications to the resource

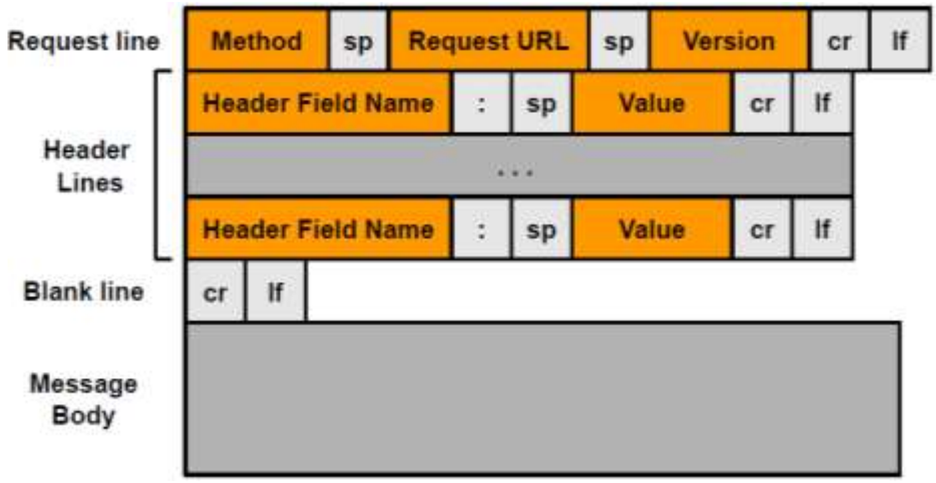


Figure 8. 2: HTTP Requests and Response

Request - URI is a Uniform Resource Identifier which identifies the resource to which the request has to be sent. HTTP Version will return the status code. Header lines tells about the specifications of content type, content length, TCP connection. HTTP Server architecture consists of HTTP Server Library, HTTP Server Sketch Sources.

HTTP Server Library,

- Implements HTTP Server framework
- ProcessServer.cpp manages Network/ WiFi connections, TCP sockets, cooperative task scheduling
- ProcessClient.cpp manages reading/ writing data from/to TCP socket, URL identification, compose functions.

- Implements some default HTML Compose Functions like ComposeHTMLSDPage() for reading HTML pages off micro SD Card
- Implements other functions also

To setup HTTP Server with FPGA, we need to configure the Static IP, Gateway IP, Subnet Mask by using Configuration and Server files as shown in the Figure 8.3. For this the particular port (port 80) of the server has to be configured as listening port of the server. Upon receiving the client request at the port 80 of the HTTP server, the server responds to the (Oh, S., 2004) request and sends back the response. The below Figure shows the HTTP Server Configuration with FPGA in which the IP address, Local Static IP, Gateway IP, Subnet mask are all set. Since we are utilizing Class C IP ware, IP address is set as '192.168.1.X'. The Gateway is set as '192.168.1.2' which tells that client address from '192.168.1.2' to '192.168.1.254' are allowed for accessing.

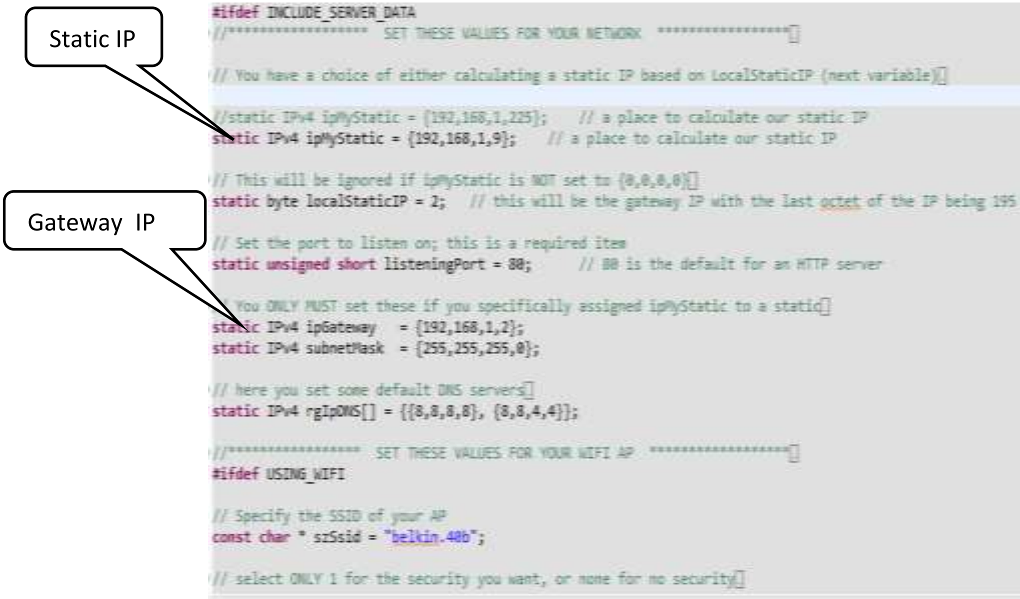


Figure 8. 3: HTTP Server Configuration

The HTTP server needs a host network to carry out the transmission of data. 'Belkin' Router is used as a router to provide the network. The Wi-Fi module will access the network of Belkin to facilitate the data transfer from FPGA to end user through Wi-Fi network.

The data transmitted should be encrypted and decrypted to avoid data hacking and leaking of data. For this WAP2 key and WEP key are used. To access the router network the user should provide SSID and password during configuration of HTTP server. Table shows the configuration of different parameters in the server.

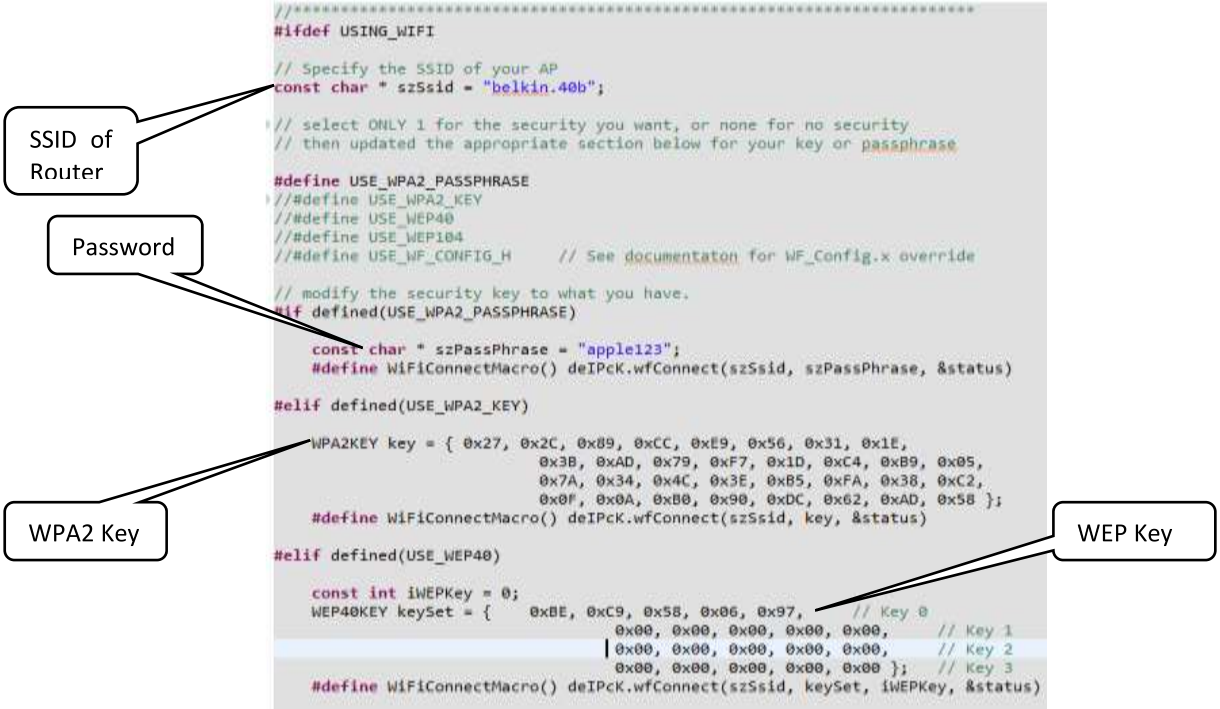


Figure 8. 4: HTTP Server Configuration

Table 8.1 shows the configuration details of the network which tells about the Static IP, Gateway IP, Subnet mask, Listening port, Login ID and password.

Table 8. 1: Configuration Details

IPv4 ipMyStatic	192.168.1.9
Listening Port	80
IPv4 ipGateway	192.168.1.2
IPv4 subnet Mask	255.255.255.0
SSID & Password	'belkin.40b' & 'apple123'
IPv4 rgIpDNS []	{{8,8,8,8},{8,8,8,4}}

The Router IP address will be the Gateway IP for the Pmod WiFi module. Router, Pmod wifi should be placed in the same network. Figure shows the Router configuration by setting up the IP address as '192.168.1.2'. Subnet Mask as '255.255.255.0', IP pool allows about 253 users to access the network. Wireless Security Settings are set with WPA/WPA2 Keys. Figure shows the Router Configuration done. Since the Router is DHCP enabled, each time the user connects the system to the network the router provides the different IP address at every time. Figure 8.4 shows the router configuration with the server.

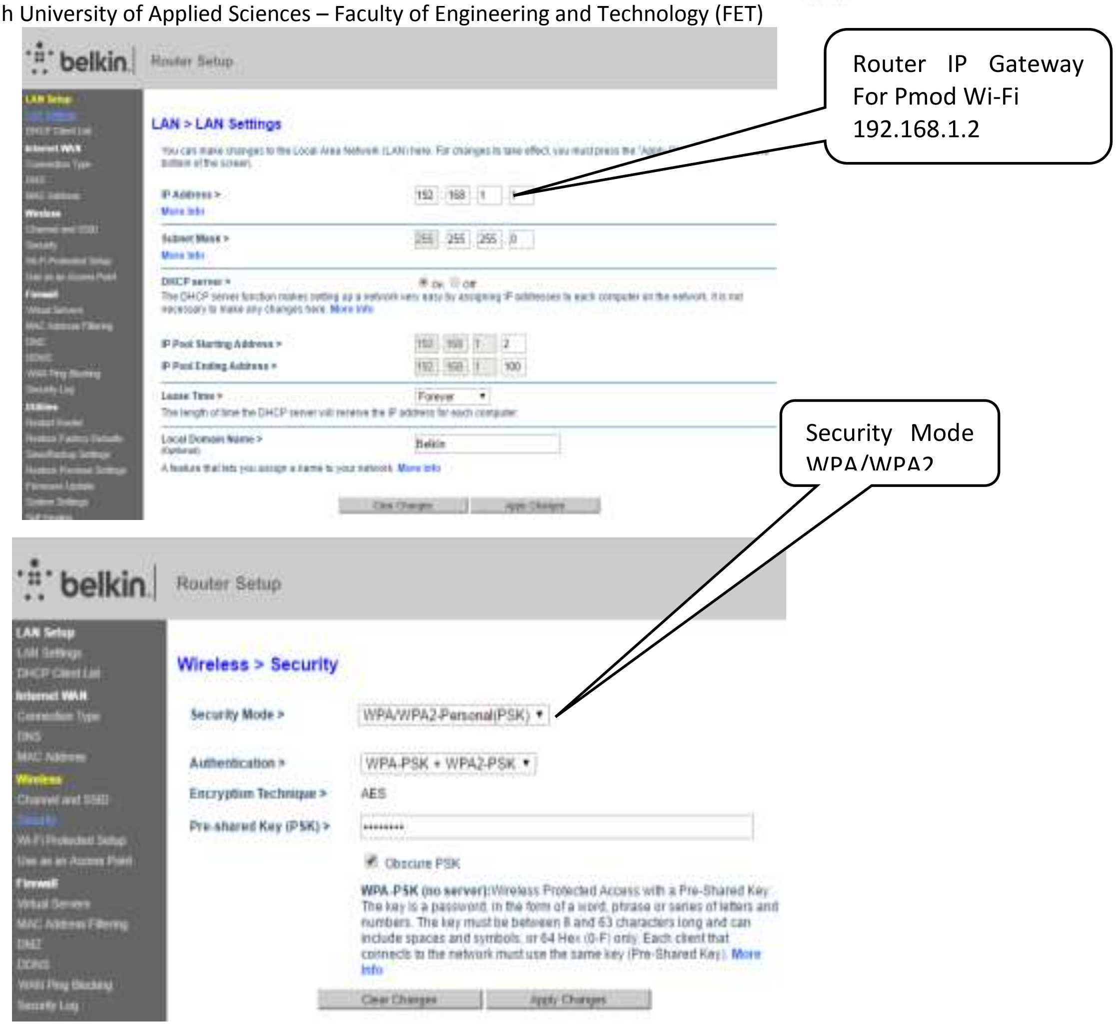


Figure 8. 5: Router Configuration

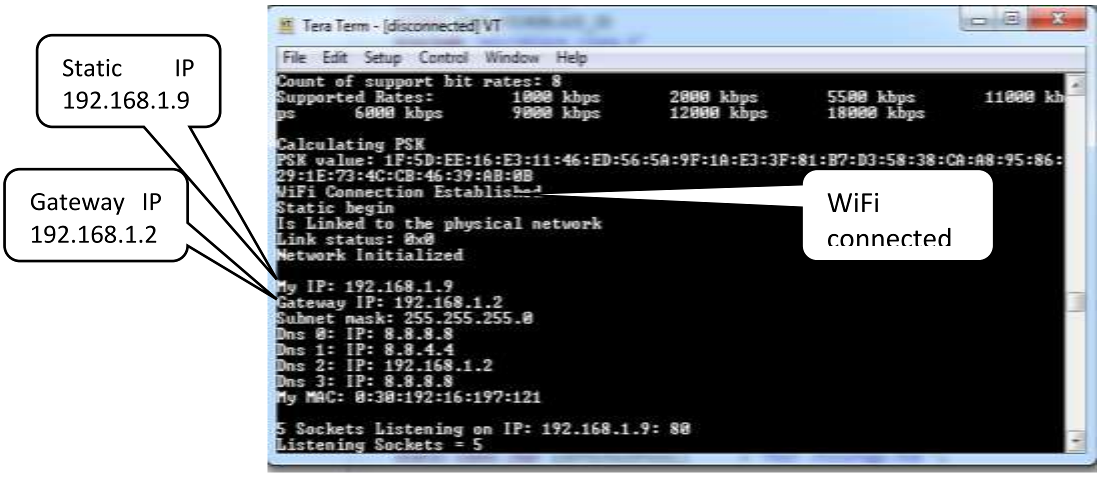


Figure 8. 6: Tera Term Status of connection to network

'Tera Term' application is open source free software which is an emulator software that gives details about serial terminal connections, USB connections and also provides status of the wifi connectivity with the network available. The Figure 8.5 shows the Tera Term status of network connected.

Micro SD Card and IP Configuration

9.1 Introduction:

The Nexys4 gives a microSD opening to both FPGA arrangement and client get to. The on-board Auxiliary Function microcontroller offers the SD card transport with the FPGA. Before the FPGA is designed the microcontroller must have entry to the SD card by means of a SPI interface. Once a bit record is downloaded to the FPGA (from any source), the microcontroller control cycles the SD space and gives up control of the transport. This empowers any SD card in the space to reset its inward state machines and boot up in SD local transport mode. The greater part of the SD sticks on the FPGA are wired to bolster full SD speeds in local interface mode as appeared in the figure. The SPI interface is additionally accessible, if necessary. When control over the SD transport is passed from the microcontroller to the FPGA, the SD_RESET flag should be effectively determined low by the FPGA to control the microSD card space. Figure 9.1 shows the Interfacing of SD Card with Artix 7.

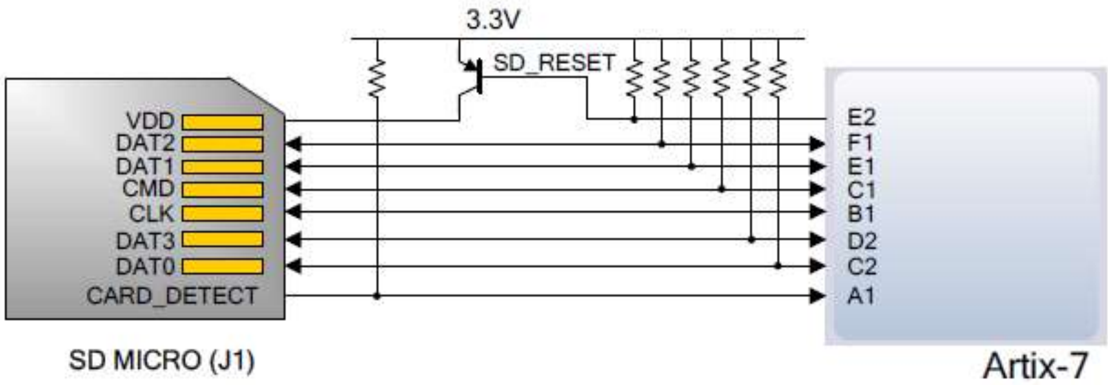


Figure 9. 1: SD Card Interfacing with Artix 7

The smaller scale SD card is connected with the web server in which the document identified with the website page is put away. Smaller scale SD card mix with MicroBlaze processor through the AXI transport originating from the AXI interconnect, with MicroBlaze as the ace, All of the SD sticks on the FPGA are wired to bolster full SD speeds in local interface mode, all html documents, photographs diagram, symbol is connected with the site pages are put away in the SD card.

The smaller scale SD space IP (Smyth, N., 2016) is designed with the HTTP Server which will permits to get to information of SD card to site page and android application, When the end client attempt to get to data of the site page then the through the web server over the Wi-Fi arrange the put away information in the SD card is connected with it.

Perusing a picture from a SD card requires a few layers of handling. The initial segment is the physical layer. The particular layouts one of two method for speaking with a cards physical layer: utilizing the SDIO convention (a four-wire, bidirectional information/order channel), and utilizing the SPI convention.

To talk to an SD card we need to implement several layers and those layers are as follows,

- Physical layer
- Data link layer
- File system layer
- Application layer

Physical layer:

SD cards bolster both a slower SPI and a quicker local SD interface. The Nexys 4 wires all SD pins to the FPGA, so you may pick whichever. Either needs a controller actualized in the FPGA that serializes yours summon and information parcels.

Data link layer:

Above the physical layer comes the command protocol implementation which implements the SD state machine. You need to issue initialization and then read/write commands specific to the SD standard.

File system layer:

The data commands read raw block/sectors (Kumar, P. et al., 2015) from the SD card. If the SD card is formatted with any file system, which it probably is, you need to implement it to have be able to navigate directories and access file.

Application layer:

In application layer we need to utilize the accompanying equipment: MicroBlaze, SPI IP. Also, programming: fatFS (open-source FAT16/FAT32 usage). FatFS calls low-level circle I/O works that you have to actualize yourself. Plate I/O for your situation will be issuing the right orders to SPI and returning information to FatFS.

The AC701 board incorporates a protected advanced information/yield (SDIO) interface to give client rationale access to universally useful nonvolatile SDIO memory cards and peripherals. The SD card opening is intended to bolster 50MHz rapid SD cards. The SDIO signs are associated with I/O bank 14, which has its VCCO set to 3.3V. Figure 9.2 demonstrates the associations of the SD card interface on the AC701 board.

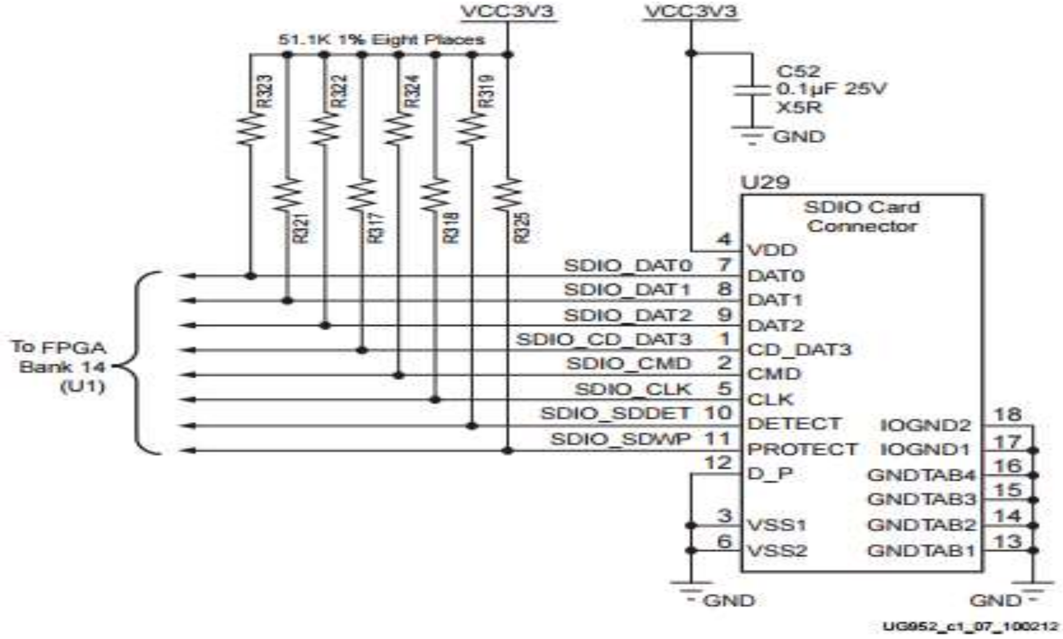


Figure 9. 2: SD Card Interface

CHAPTER 10

Web Page Development

10.1 Objectives:

- Use a text editor to author an HTML document
- Be able to use basic tags to denote paragraphs, emphasis or special type
- Create hyperlinks to other documents
- Add images to your document
- Use a Graph to Display Sensor Data

10.2 HTML Files:

HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page
- An HTML file must have an htm or html file extension

The minimum required information for a web document and all web documents should contain these basic components. The first tag in your html document is <html>. This tag tells your browser that this is the start of an html document. (Sha-Sha Shelyna , S., 2015) The last tag in your document is </html>. This tag tells your browser that this is the end of the html document.

The text between the <head> tag and the </head> tag is header information. Header information is not displayed in the browser window.

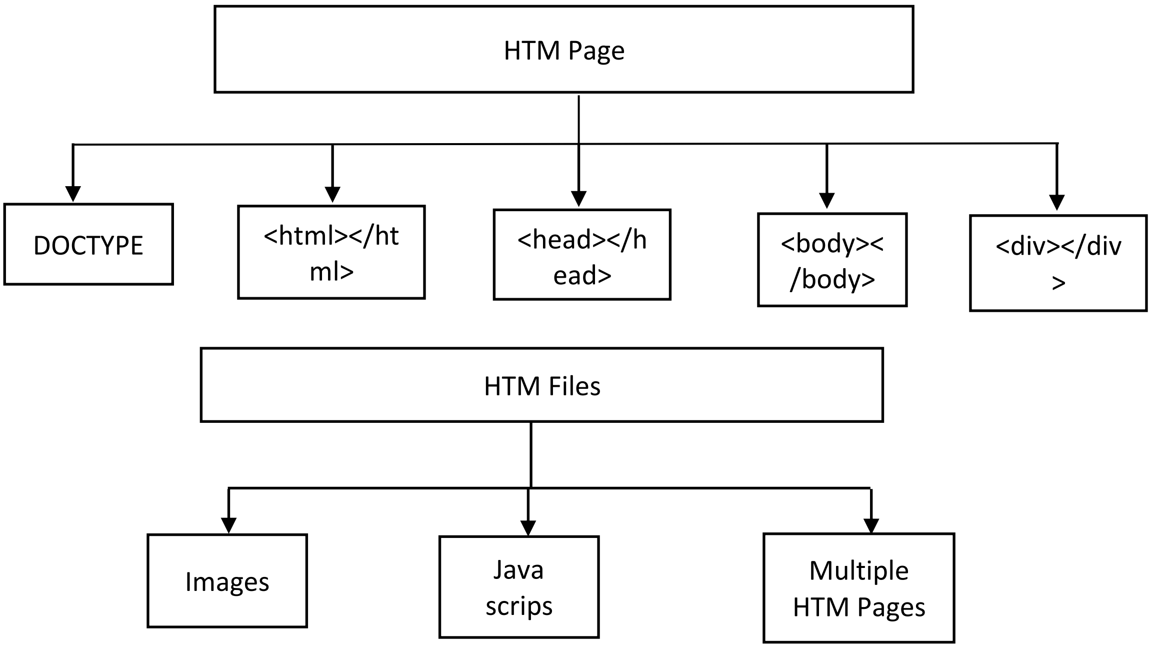
The text between the <title> tags is the title of your document. The <title> tag is used to uniquely identify each document and is also displayed in the title bar of the browser window.

The text between the <body> tags is the text that will be displayed in your browser.

What are HTML tags?

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < h1>
- The surrounding characters are called angle brackets
- HTML tags normally come in pairs like <p> and </p>
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start and end tags is the element content

- HTML tags are not case sensitive, <p> means the same as <P>



Home page:

The Web page developed has four HTML links which redirects the user to the different pages upon clicking them. Figure 10.1 shows the home page of the website of IP address 192.168.1.9. The content required can be in the body of the html code, same will be displayed in the webpage.

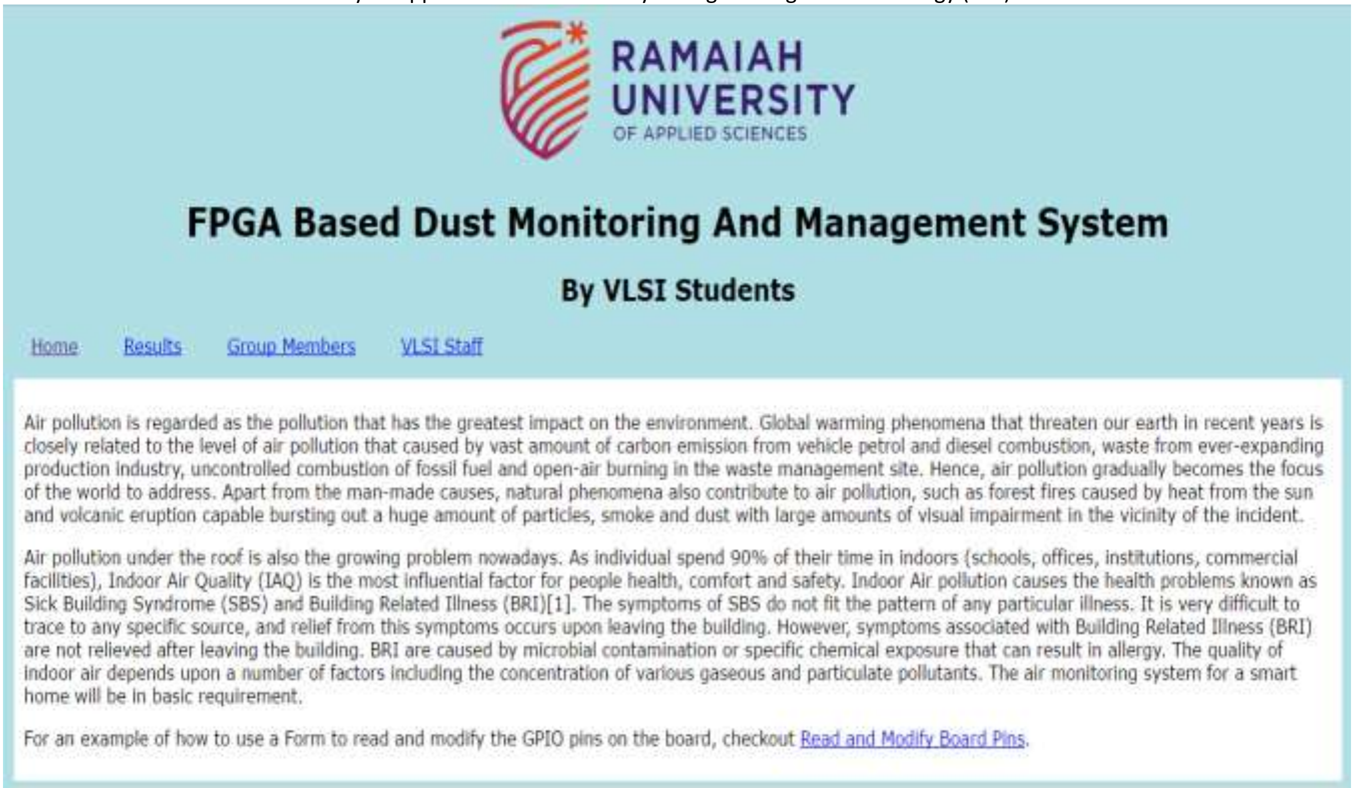


Figure 10. 1: Homepage of the Dust Monitoring Website

Graph Page:

To display the concentration of dust present in the environment, a graph is developed in the webpage which tells the range of values of dust present at the sensor input. The meter indicates good, moderate, severe conditions of the dust values. Figure 10.2 shows the designed graph for displaying the values in mg/m^3 scale.

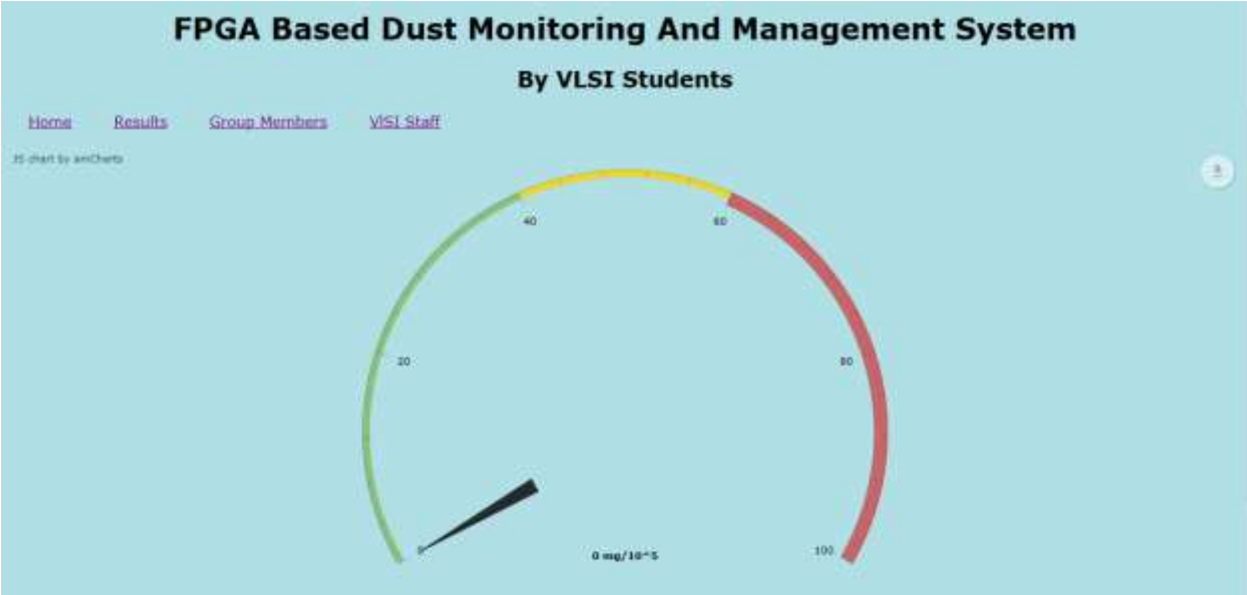


Figure 10. 2: Graph created to display the concentration of dust

---

Android Application Development

11.1 Introduction:

Android is the name of the operating system for the hand held devices, developed by well-known company “Google”. It has been more than nine years since the launch of Android globally. There are different versions of Android like the different platforms for accessing higher version applications. Majorly, the applications will be available on PLAY STORE or as in the form of android application package majorly knows as .apk file. The android version started from 1.5 which is cupcake to the latest 7.1 knows as Nougat.

The android smart phones (Ricardo J., Helena S.,2016) are used in mass for hand held devices, hence choosing the android platform is vital so as to determine the contents of the dust particle in the atmosphere present as more number of people is aware of the smart phone applications and their graphical user interface.

The android application is developed using the Android development kit known as Android Eclipse. For developing any android based application, Java development kit is required which is the first step in developing the android application. The android application can be developed on any operating system. Starting of the android application require adding of the packages like support tools, build tools, android support library, Google USB driver. The Android SDK can be downloaded for the different versions required starting from API 14 to API 21. Figure 11.1 shows Android SDK files for android application development

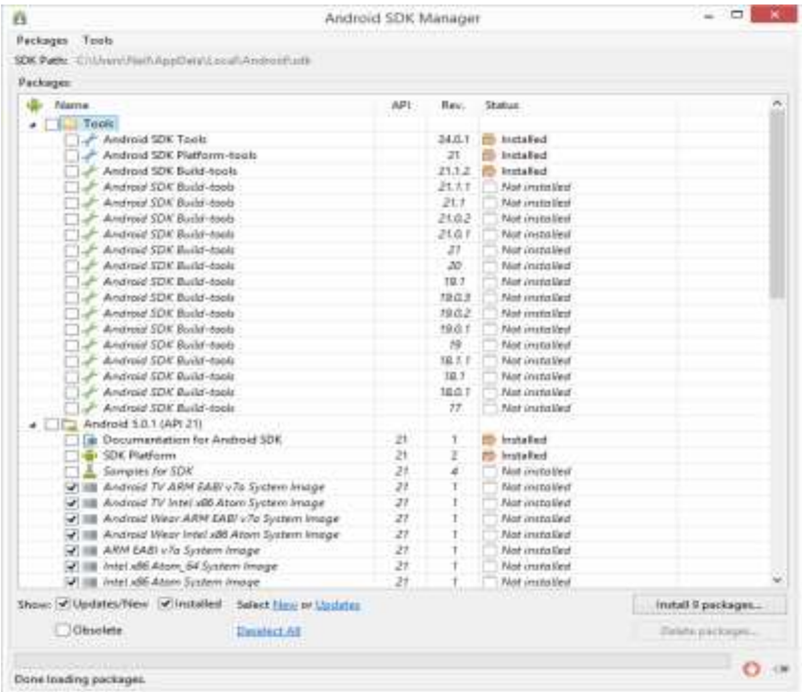


Figure 11. 1: Installing the Android SDK files

To start with the application name is added along with the company domain name which could be anything. The package name has to be unique for every application else the play store wont accept as it clashes with the already existing appplication. The activity has to be chosen which is the background for the application. Figure 11.2 shows the application name and Package name

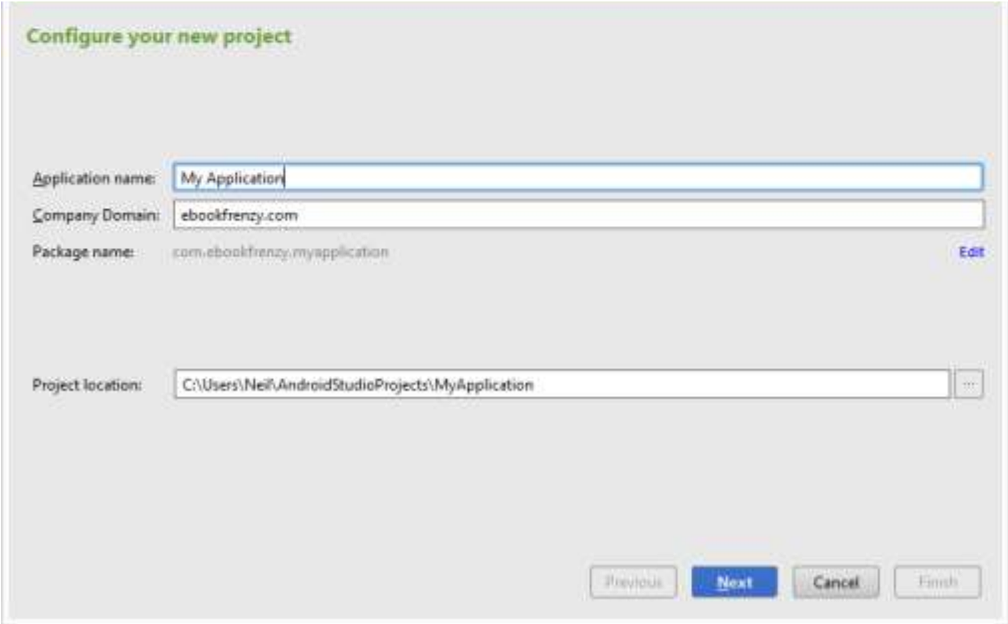


Figure 11. 2: Creating the Application

The activity is based on the requirement of the application, majorly the blank activity is chosen for the development, for developing the location based application the desired activity is chosen from the activity. For developing the Dust monitor application the blank activity is chosen. Figure 11.3 shows the activity selection for the application.

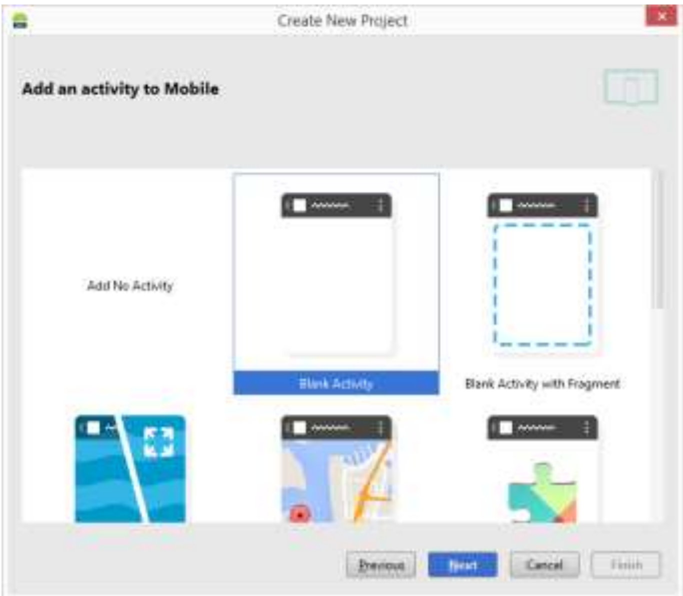


Figure 11. 3: Selecting the Required Activity

Once created the android development page which shows the source file and android manifest.xml file. The “hello world” appears on the screen. The text can be changed selecting the Palette, the buttons are added choosing from the palette. In the developed application, the refresh button is added in the widget panel and the data is refreshed every time the refresh button is pressed. The best feature of the Eclipse is the changes which are made to the XML layout the same will be featured on the preview panel. Figure 11.4 shows Android eclipse page for application development

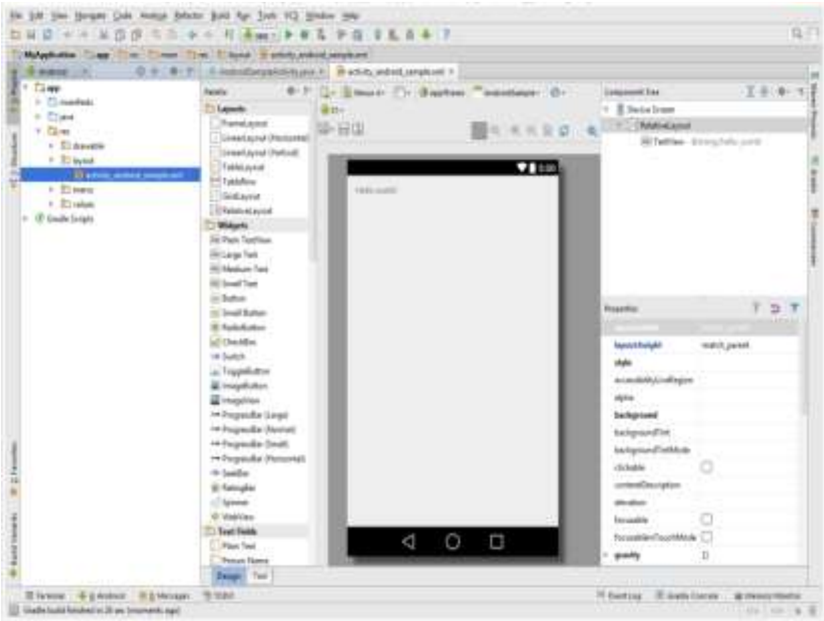


Figure 11. 4: Opening the Android Eclipse Displaying the Activity

The right hand side Project tool Window contains main features which are :

src

This contains the **.java** source files for the developed project. By default, it includes a MainActivity.javasource file having an activity class

gen

This contains the **.R** file, a compiler-generated file that references all the resources found in the project. This should not be modified.

bin

This folder contains the Android package files **.apk**built by the ADT during the build process

res

This is a directory for drawable objects that are designed for high density Screens.

layout

This is a directory for files that define your app's user interface.

values

This is a directory for other various XML files that contain a collection of resources, such as strings and colors definitions.

AndroidManifest.xml

This is the manifest file which describes the fundamental characteristics of the app and defines each of its components.

The Logo for the application is chosen as the Logo of the "Ramaiah University". Figure 11.5 shows the Logo for the application.

Figure 11. 5: Ramaiah University Logo

Android virtual device are essentially emulators that allow Android applications to be tested without the necessity to install the application on a physical Android based device. An Android virtual device may be configured to emulate a variety of hardware features including options such as screen size, memory capacity and the presence or otherwise of features such as a camera, GPS navigation. When launched, an Android virtual device will appear as a window containing an emulated Android device environment. New Android virtual devices are created and managed using the Android Virtual Device Manager. Figure 11.6 shows different classes of java.

```
package com.example.speedmeeter;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;

import android.app.Activity;
import android.content.res.Resources.NotFoundException;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity
{
        Handler mHandler;
        private static final String TAG = "MainActivity";
        TextView txtdate;
        /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
```

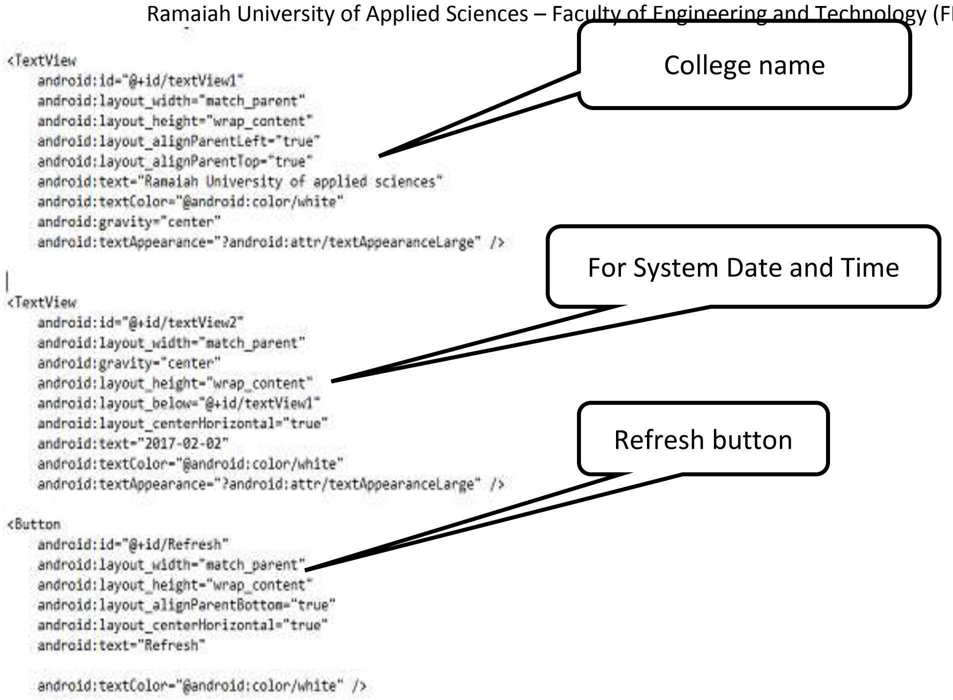
Figure 11. 6: Importing the Different Classes of Java

For the developed application the package name is given as the Speedometer and the functions are imported as shown in the above figure like the simple date format which shows the current date and time of the system i.e., the time and date are fetched from the mobile and the same will be displayed. The android.widget.button showing the refresh button and displaying whenever the refresh button is pressed, it displays that "clicked on refresh". Handlers are bound with the functions and helps in continuously displaying without being going out of the loop. Importing text view which is imports the texts which are written in Android.manifest.xml file. Figure 11.7 displays the refresh button, data and time settings.

```
@Override
public void onStart(){
    super.onStart();
    final Speedometer speedometer = (Speedometer) findViewById(R.id.speedometer);
    Button increaseSpeed = (Button) findViewById(R.id.IncreaseSpeed);
    Button decreaseSpeed = (Button) findViewById(R.id.DecreaseSpeed);
    increaseSpeed.setOnClickListener(new OnClickListener(){

                    @Override
                    public void onClick(View view) {
                            speedometer.onSpeedChanged(speedometer.getCurrentSpeed()+10);
                    }

    });
    decreaseSpeed.setOnClickListener(new OnClickListener(){

                    @Override
                    public void onClick(View v) {
                            speedometer.onSpeedChanged(speedometer.getCurrentSpeed()-10);
                    }
    });

    Button refresh = (Button) findViewById(R.id.Refresh);
    refresh.setOnClickListener(new OnClickListener() {

                    @Override
                    public void onClick(View v) {

                            speedometer.onSpeedChanged(0);
                            txtdate.setText(new SimpleDateFormat("dd MMM yyyy h:mm a",Locale.US).format(new Date()));
                            m=0;

                    }
            });
    triggerEvery3Sec();
```

Figure 11. 7: Functioning of Refresh Button, Date and Time

The increase and the decrease button show the increase of the data received or the decrease in the data. The format for the date and time is as shown above. The speedometer is developed by importing the speedometer library and the changing the value to 0 to 100 as per the requirement and the remaining region is colored with black as only the upper eclipse is required for the data displaying. The region of the data indicator is colored with orange and displays the data whenever the data us received as it triggers for every 3seconds which is given in the socket time. Figure 11.8 shows the android manifest file for representing the date, time, college name, refresh button.



```
<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:text="Ramaiah University of applied sciences"
    android:textColor="@android:color/white"
    android:gravity="center"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:gravity="center"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:text="2017-02-02"
    android:textColor="@android:color/white"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<Button
    android:id="@+id/Refresh"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:text="Refresh"

    android:textColor="@android:color/white" />
```


Figure 11. 8: Android Manifest Showing the Text View of College Name, Date and Time, Refresh Button

The display of the text view for displaying the college name, its apperances in colour and orientation, likewise for the system date and time and for the refresh button. Match_parent showing that the matching with the entire width or length as required.The wrap_content is it wraps the data within the required space. Android:gravity shows the appearnces of the text or button on the application.The android. Manifest is like bone, skin and structure where as Java is heart and brain in simple words. Figure 11.9 shows the IP address pinging.

```
private String LightsOnOff(String 192.168.1.9) throws Exception

 {

    StringBuffer response = null;

    try

    {

       192.168.1.9 obj = new 192.168.1.9(192.168.1.9);

       Http192.168.1.9Connection con = (Http192.168.1.9Connection) obj.openConnection();

       con.setRequestMethod("GET");

       con.setRequestProperty("User-Agent", "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.4; en-
US; rv:1.9.2.2) Gecko/20100316 Firefox/3.6.2");

       BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));

       String inputLine;

       response = new StringBuffer();

       while ((inputLine = in.readLine()) != null)

       {

          response.append(inputLine);

       }

       in.close();
```

Figure 11. 9: Java Code for Pinging the IP Address

The http server gets the data from the Pmod WiFi device data after sensing the dust data. The same data will be received by http server. The information can be accessed from the server with the help of IP address. The Android device pings the http server every 3 seconds as defined in the socket time. The procedure is as follows, the open.get is the first step where the connection will be open for the data to receive, the input stream reader is the one which differentiates the junk data from the original data. The final step is the buffer reader which passes the information to the speedometer whenever the IP address is pinged and the same information will be displayed on the developed android application. The http.h file is used to get the response from the server. Figure 11.10 shows the developed android application on Android phone.

Figure 11. 10: The Developed Application

The developed application looks like this without the data. For the developed application the given name is Dust monitor. When the refresh is pressed the data will reset. When the device is connected with the dust sensor and the Pmod WiFi is throwing the data out then the information can be accessed by connecting the device with the internet and the data can be seen on the application developed. The IP address required for the developed application is the Class C IP address which has 3 network and 1 host address which is easier for private applications. For sending the application to the phone, the phone has to be in developer mode only then the application can be sent to the phone. For creating the application package (.apk file)there are two ways which is signed apk and unsigned apk. For signed apk the code has to be generated and then it can be uploaded to the Play store by paying money. The developed application is unsigned application package and can be sent to any phone if it is in developed mode. For making the phone in developer mode, in settings, the kernel option has to be pressed 7 times thus enabling the device in developer mode. The developed android application can be accessed on Android 4.0 and above. Figure 11.11 shows the application developed on the smartphone.

Figure 11. 11: Application on Smartphone

The above figure shows real time displaying of the developed application on android phone with the logo

Figure 11. 12: Good Dust Density Values

Figure 11.12 shows the good range of dust density. As defined above the dust density values distinguishing between good, moderate and bad dust values. The application is showing for different dust values present.

Figure 11. 13: Moderate Dust Density Value

The figure 11.13 shows Moderate dust density value present as classified in the above mentioned dust density value.

Conclusion and Future suggestion

Conclusion:

The sensor used is SHARP compact dust sensor, whose behavioral model is analyzed in MultiSim and found that the processor should take the sample values with the delay of 0.2ms-0.3ms also found that the sampling period of sensor values are 0.02ms-0.04ms. The sensor analog values are converted into Digital values by processing the values through the JXADC port of the Nexys-4 Artix-7 board. The Pmod Wi-Fi device is configured using the Vivado 16.3 and the Micro Blaze Software Development Kit, the board takes up 2 clock cycles one for the board and other for the Wi-Fi device operating at 100MHz and 25MHz respectively. The power consumed by the device is found to be 0.356 watt and found Zero Slack. The HTTP.h header file is used for configuring the web page, to connect with the web page and displaying all the contents of the web page. The SSID and password of the router will be given in the HTTP.h file, to connect with the router. Configuration of the router for accessing the information via internet. The static IP address is defined and the IP address used is private IP address the IP configured is 192.168.1.9. Once the entire above configuration is done there will be connection established between the FPGA board and http server which sets the path for data transmission. The html file is developed for displaying the values of the dust on the web page, the developed web page is developed using HTM format. Head and body are important for displaying the contents in the web page when the values are received. Graph is developed for displaying the value using the java script. When the system is connected with the IP address and linked to the static IP address as given in HTTP.h file then the web page starts showing the dust monitor values. The Android application is developed using the Android Eclipse for displaying the values of the dust sensor on the hand held device. As android is most widely used operating system for hand held device worldwide. The developed application is compatible with the android version 4.0 and above. The speedometer is developed for displaying the values of the dust sensor, when the device is connected with the specified IP address via router then the system pings to the IP address for every 3 second as defined in the socket time and receives the data and the same is displayed on the application indicating the dust density.

Future suggestion:

The sensor considered here is low resolution sensor and the sensor can be increased to higher resolution, for detecting minute dust particles with less voltage consumption. The Wi-Fi has the limitation of 400 meter range, hence need the Ethernet cable to be connected and monitoring these at the remote places is difficult as they need the Ethernet cables and Wi-Fi router, hence LORA can be implemented for long distance monitoring, enabling up to 1KM range. The access points can be used to increase the range of the Wi-Fi. The developed Android application only displays the values of the dust sensor, the work can be extended so that the application can alert the user by sending the notification, when the dust density value crosses the safe zone. The Web page displays the values and the color of the developed application and the meter range varies for different values, the same can be extended so that it can alert the user by giving the beep sound when the value crosses the safe zone.

Digilent, (2013) Nexys4™ FPGA Board Reference Manual, Henley Court Pullman, WA.

Frost, R. (2013) Applied Kinesiology, Revised Edition: A Training Manual and Reference Book of Basic Principles and Practices. North Atlantic Books.

Kumar, P. et al., 2015. The rise of low-cost sensing for managing air pollution in cities. *Environment International*, 75, pp.199–205. Available at: http://dx.doi.org/10.1016/j.envint.2014.11.019.

Norhafizah, R. (2015) Air Pollution Index Real Time Monitoring System, Master Thesis, University Tun Hussein Onn Malaysia.

Oh, S., (2004). Room environment monitoring system from PDA terminal. *Proceedings of 2004 International Symposium on Intelligent Signal Processing and Communication Systems, 2004. ISPACS 2004.*, 2, pp.497–501.

Ricardo J., Helena S. (2016) A Wireless Biosignal Measurement System using a SoC FPGA and Bluetooth Low Energy, International Conference on Consumer Electronics-Berlin, pp. 36-40.

Sha-Sha Shelyna , S. (2015) Real Time Air Quality Reporting System, Master Thesis, University Tun Hussein Onn Malaysia.

SHARP, Application note of Sharp dust sensor GP2Y1010AU0F - Sheet No.: OPI3024EN. , pp.0–6.

Sources, A.E., (2012) Particulate Matter. , (March), pp.7–9.

Smyth, N. (2016) Android Studio 2 Development Essentials. eBookFrenzy.

World Health Organization, 2013. Health Effects of Particulate Matter: Policy implications for countries in eastern Europe, Caucasus and central Asia. *Journal of the Korean Medical Association*,50(2),p.20.