JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 892 - 899
https://publishoa.com
ISSN: 1309-3452

Power Efficient Air Purifier Employing PM2.5 Air Dust Monitor Employing Embedded Module

M. Mahesh Kumar[1], S.Abdul Malik[2], Syed Munawwar[3]

[1]Department of ECE, Santhiram Engineering College, Nandyal, India,
[2]Department of ECE, Ashoka Women's Engineering College, Kurnool, India.
[3]Department of ECE, Santhiram Engineering College, Nandyal,India.

ABSTRACT

As air pollution is on the rise, providing cleaner air inside building is becoming an increasing demand. In this project, a PM 2.5 Air Dust sensor is employed to measure the concentration of minuscule dust particles in the air. The measured value is given as an input to the embedded module. The embedded module based on the magnitude of the measured value gives the control signal to the blower fan motor of the air purifier in terms of how fast it should rotate. This is a power efficient way to purify the air. When the concentration of dust is low, there is no need for the blower fan of the air purifier to be fast and send a high volume of air to be purified.

Keywords: **Embedded, Sensors, Arduino, Air Purifier**

1. Introduction

The current pandemic situation has shown that a great deal of precaution should be taken against air borne diseases like corona and black fungus. Inside houses and buildings, the accumulation of dust particles is more since the concentration of this is high. The furniture and other things physically occupying the room become either a source or an accumulating point of fine dust particles. There are current systems that identify the PPM dust level in the room and initiate filtering activity accordingly. These filters generally absorb the air in the room by fan suction, pass it through filters and then send it out again into the room. In this paper, a simple incremental innovation is presented. The suction of air volume is controlled by the speed of fan's rotation by the sensor input obtained from PPM dust sensor. That is the fan's rotary speed and correspondingly the volume of the suction air. This idea is for two-fold benefit. First, reducing the fan's speed for low PPM dust levels conserves electric power. Secondly, very low PPM dust levels are not accepted as healthy since a certain level of minimal dust particles are required to develop respiratory immunity.

2. Proposed System

The system is for providing a purified environment expelling the unwanted gases and micro particles. The system consists of a gas sensor, Arduino UNO, relay and a DC Fan. The system senses for the presence of these harmful gases and the data is sent to the Arduino UNO and the fan runs. The stepwise process is explained in further briefing. For purifying we use MQ135 gas sensor senses gases like ammonia nitrogen, oxygen, alcohols, aromatic compounds, sulphide, and smoke. The boost converter of the chip MQ-3 gas sensor is PT1301. The operating voltage of this gas sensor is from 2.5V to 5.0V. It has the potential to detect different harmful gases. The threshold value of particulate matter is set. When the pm value is low than threshold, the fan doesn't spin. When the pm value is more than threshold, the Arduino is coded so that is enables the fan to spin.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 892 - 899
https://publishoa.com
ISSN: 1309-3452

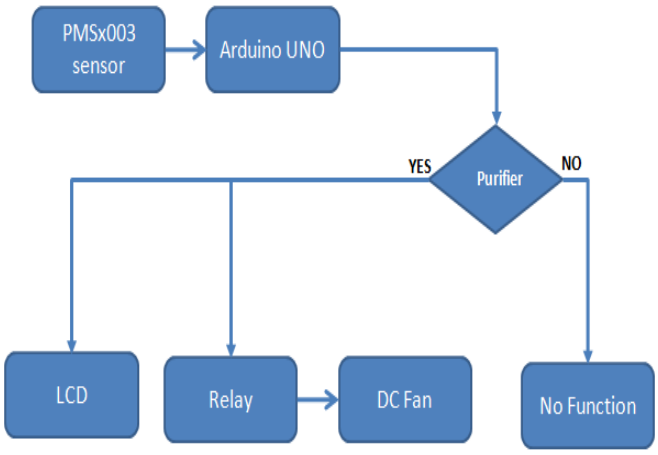

Figure 1. Block Diagram

3. Arduino Microcontroller

"The microcontroller used to implement this project and similar projects are Arduino boards. Presented below is an outline of Arduino family boards."[2]

3.1 Introduction

"Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. It's hardware products are licensed under a CC-BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors"[2]

"Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards ('shields') or breadboards (for prototyping) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs. The microcontrollers can be programmed using the C and C++ programming languages, using a standard API which is also known as the Arduino language. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) and a command line tool (Arduino-cli) developed in Go"[2]

"The Arduino project began in 2005 as a tool for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats and motion detectors. The name Arduino comes from a bar in Ivrea, Italy, where some of the founders of the project used to meet. The bar was named after Arduino of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014"[2]

3.2 Hardware

"Arduino is open-source hardware. The hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino website. Layout and production files for some versions of the hardware are also available"[2]

"Although the hardware and software designs are freely available under copy left licenses, the developers have requested the name Arduino to be exclusive to the official product and not be used for derived works without permission. The official policy document on use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the project name by using various names ending in –duino"[2]

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 892 - 899
https://publishoa.com
ISSN: 1309-3452

"An early Arduino board with an RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are at the top, the 6 analog input pins at the lower right, and the power connector at the lower left."[2]

"Most Arduino boards consist of an Atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, or ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino Due, based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed shields. Multiple and possibly stacked shields may be individually addressable via an I²C serial bus. Most boards include a 5 V linear regulator and a 16 MHz crystal oscillator or ceramic resonator. Some designs, such as the Lily Pad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions."[2]

"Arduino microcontrollers are pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory. The default boot loader of the Arduino Uno is the Optibootbootloader. Boards are loaded with program code via a serial connection to another computer. Some serial Arduino boards contain a level shifter circuit to convert between RS-232 logic levels and transistor–transistor logic (TTL) level signals. Current Arduino boards are programmed via Universal Serial Bus (USB), implemented using USB-to-serial adapter chips such as the FTDI FT232. Some boards, such as later-model Uno boards, substitute the FTDI chip with a separate AVR chip containing USB-to-serial firmware, which is reprogrammable via its own ICSP header. Other variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. When used with traditional microcontroller tools, instead of the Arduino IDE, standard AVR in-system programming (ISP) programming is used. An official Arduino Uno R2 with descriptions of the I/O locations. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.1-inch (2.54 mm) headers. Several plug-in application shields are also commercially available. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboards."[2]

"Many Arduino-compatible and Arduino-derived boards exist. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education, to simplify making buggies and small robots. Others are electrically equivalent, but change the form factor, sometimes retaining compatibility with shields, sometimes not. Some variants use different processors, of varying compatibility."[2]

3.3 Software

"A program for Arduino hardware may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit AVR and 32-bit ARM Cortex-M based microcontrollers: AVR Studio (older) and Atmel Studio (newer)."[2]

IDE:

"The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, and Linux) that is written in the Java programming language. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2."[2]

"The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU tool chain, also included

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 892 - 899
https://publishoa.com
ISSN: 1309-3452

with the IDE distribution. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware."[2]

Pro IDE:

"On October 18, 2019, Arduino Pro IDE (alpha preview) was released. The system still uses Arduino CLI (Command Line Interface), but improvements include a more professional development environment, auto completion support, and Git integration. The application frontend is based on the Eclipse Theia Open Source IDE. The main features available in the alpha release are:"[2]

Sketch:

"A sketch is a program written with the Arduino IDE.[64] Sketches are saved on the development computer as text files

with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension."[2]

"A minimal Arduino C/C++ program consists of only two functions:

setup( ):

This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch. It is analogous to the function main ().

loop ( ):

After setup () function exits (ends), the loop() function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset. It is analogous to the function while."[2]

4. Industry 4.0

"Wikipedia defines Industry 4.0 as thus: The Fourth Industrial Revolution (4IR or Industry 4.0) is the ongoing automation of traditional manufacturing and industrial practices, using modern smart technology. Large-scale machine-to-machine communication (M2M) and the internet of things (IoT) are integrated for increased automation, improved communication and self-monitoring, and production of smart machines that can analyze and diagnose issues without the need for human intervention."[3]

Automation under Industry 4.0 has a particular schema or pattern at its outset. Presented below is how automation in the mass production industry as well as consumer level products are built in today's technological era.

The schema presented in Figure 2 has a lot of other components involved but the generic outline of it stands justifiable for all kinds of automation today.

The software automaton of the conventional automation model, which is the status quo, was built by a human expert or a team of human experts till now. With the advent of machine learning technology, the software automaton was not fully directly designed by human experts. The human experts build the machine learning software and give the real world data set as training information. The machine learning software identifies the pattern between the input and the output parameters of the dataset in the form of a mathematical model. This mathematical model can be downloaded as a working software module to other electronic computing devices. This mathematical model is referred to as the 'trained machine learning module'. The software automaton of all the current digital embedded devices is a mathematical model that gives a numerical output for a numerical input based on arithmetic and logical conditions. This software automaton, as explained above can be either directly developed by a set of human experts by means of setting the boundary conditions themselves based on observation and requirement or can be downloaded as an executable module from machine learning training systems that are trained with relevant dataset. In whatever way the software automaton is developed, it can be loaded onto the relevant embedded computing module that can be used for either sensor based closed loop automation or open loop automation.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 892 - 899
https://publishoa.com
ISSN: 1309-3452

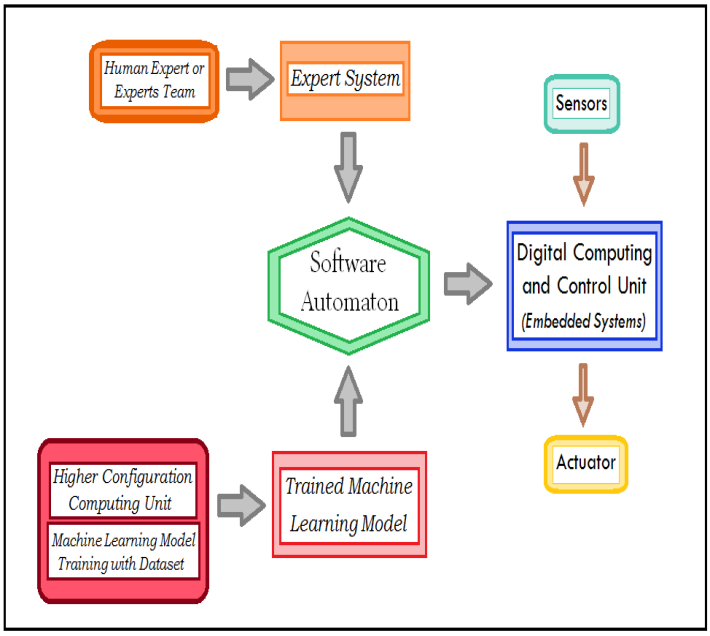

Figure 2. Schema of Automation

The technological components of Industry 4.0 includes IoT, augmented reality, virtual reality, cloud computing, 3D printing, big data analytics, networking, data security, human-machine interaction and others. IoT is a very effective way to collect real world data. Sensors integrated with data acquisition and transmission systems can be placed anywhere and the collected data can be pre-processed if required and used as datasets to train machine learning models.

Cloud computing is employed for optimized utilization of computing resources. There are many third party vendors like Google and Amazon which are very reliable in terms of data security and speed of computation. These services offer companies and organizations a cheap and reliable way to harness the power of artificial intelligence and machine learning.

Big data analytics is the set of technological components involved with collecting, collating and managing large quantities of data for analytics and decision making. When so much data is involved, especially with third party service providers, data security plays an important role.

One of the paramount concerns about Industry 4.0 is the unemployment it can create due to powerful automations. The field of human-machine interactions and co-working has been a very developing field now to mitigate the above mentioned problem.

5. Working

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 892 - 899
https://publishoa.com
ISSN: 1309-3452

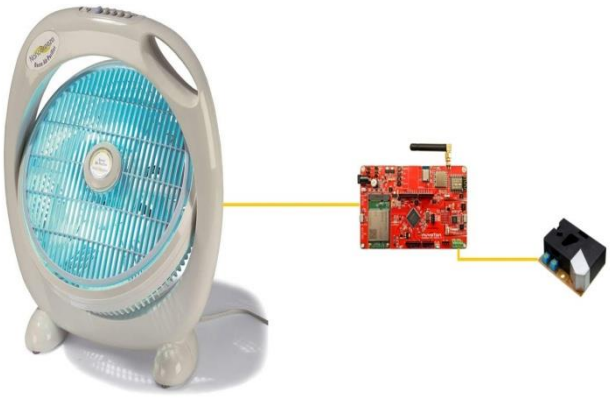

Figure 3. Schematic Diagram

Presented above is a schematic diagram (Figure 3) of a dust particle sensor integrated with a microcontroller, which is in turn connected to an air purification fan. The amount of air sucked by this fan will be proportional to the dust particles measured in the air. The amount of dust sensed in the air is continuously updated to the server via IoT. A lot of data analytics can be performed in that data.

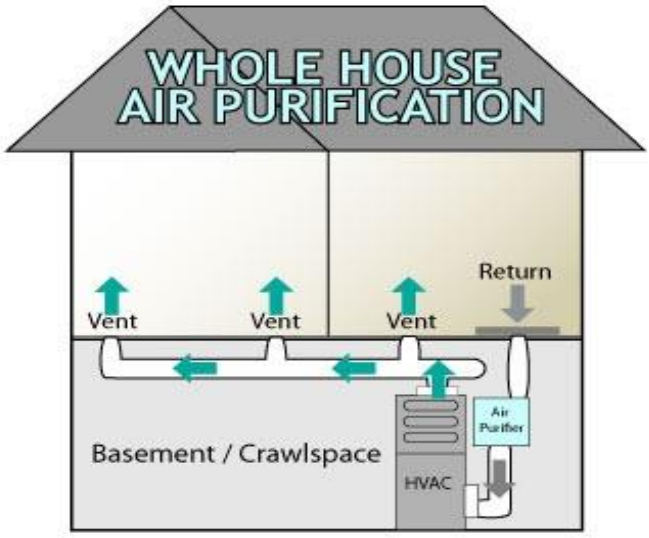

Figure 4. Air Purification System 1

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 1, 2022, p. 892 - 899
https://publishoa.com
ISSN: 1309-3452

Figure 5. Air Purification System 2

Presented above are two images (Figure 4 and 5) of schematic representation of status quo air purification systems.

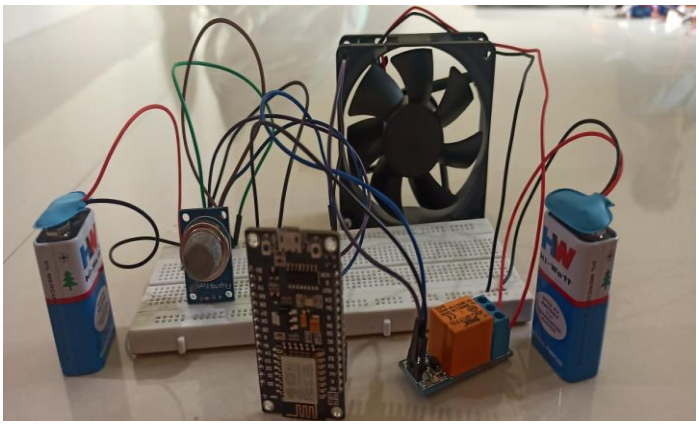

Figure 6. Android Setup

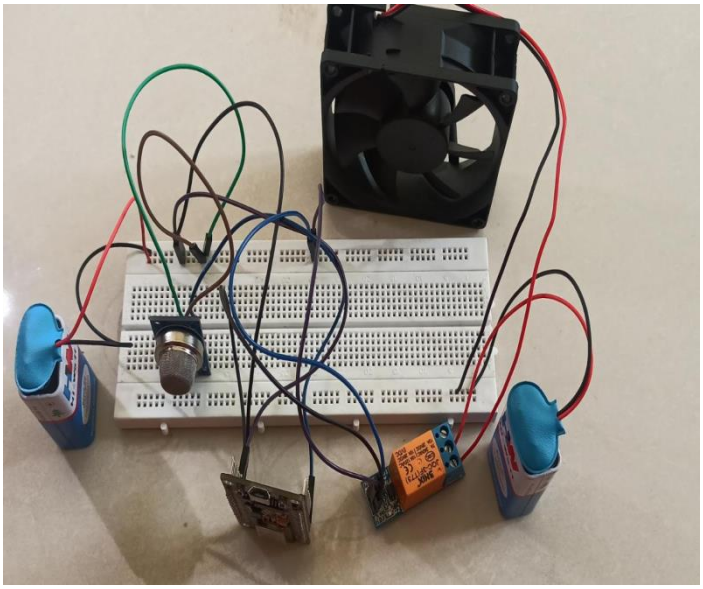

Figure 7. Output Image Top view

As shown in the Figure 6 and Figure 7, the circuit of the system consists of a PPM air dust sensor, an Arduino Uno board and a suction fan for pumping air into the air filter. This is an ideation level prototype of the product presented in this paper. The industrial grade prototype of this product will have a very similar setup as above. Multiple sensor data fusion can be implemented for high level accuracy and reliability of the system. The difference will be that more industrial grade components will be used. Standardized tabulations for a city's dust levels at various places can be generated employing this system. This circuitry was tested and the performance was satisfactory in terms of validating the proposed method. In the industrial grade prototype this system can be integrated with an IoT module and the data will be transmitted to the cloud computing enabled server. Analytics can be performed on the collected, collated and pre-processed data for obtaining useful inferences of the system and the environment.

6. Conclusion

The project mentioned here has this simple PPM sensor output based on fan speed control. It should be observed that the correlation or mapping between the volume of air to be purified is not mapped in this project. This is a potential area of research pertaining to this paper. There can be a linear smooth mapping between the PPM measurement and the volume of air suction into the filter. This can be achieved by mapping the sensor outputs and the corresponding air flow required at periodic intervals. For instance, a simple XY graph plot of PPM counts on the X axis and the corresponding

air flow volume for them at Y axis for ten values is sufficient to complete a linear regression machine learning training model. Then the values of PPM can be given as an input and the corresponding air flow can be obtained by controlling the RPM of the fans. Varying air filtering technologies include HEPA style filtration, activated carbon, ultraviolet light and ionizers. The most appropriate filtering air volume flow for PPM levels will vary for these different filtering mechanisms. When linear regression is applied, the output for different filters will be lines with varying slopes and y-intercepts.

References

[1]. Guoliang Liu, Manxuan Xiao, Xingxing Zhang and Csilla V Gal, "A review of air filtration technologies for sustainable and healthy building ventilation", Sustainable Cities and Society, April 2017.
[2]. https://en.wikipedia.org/wiki/Arduino
[3]. https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution