VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi – 590018, Karnataka.

2023-2024

A Project report on

"NODE MCU BASED AIR AND NOISE MONITORING SYSTEM OVER IoT"

Submitted in partial fulfilment of the requirements for the award of degree

BACHELOR OF ENGINEERING

IN

ELECTRONICS & COMMUNICATION ENGINEERING

BY

ESHWAR	**(3RB20EC010)**
MAHADEV	**(3RB20EC017)**
MOUNESHWAR	**(3RB20EC019)**
PRASAD	**(3RB20EC022)**

Under the Guidance of

Dr. Hameed Miyan

Department of E&CE

BHEEMANNA KHANDRE INSTITUTE OF TECHNOLOGY

Department of Electronics & Communication Engineering

Bhalki -585328

S. V. E. Society's

Bheemanna Khandre Institute of Technology, Bhalki-585328

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the Project entitled **"NODE MCU BASED AIR AND NOISE MONITORING SYSTEM OVER IoT"** *has been successfully carried out by* **Eshwar (3RB20EC010), Mahadev (3RB20EC017), Mouneshwar (3RB20EC019) & Prasad (3RB20EC022)** *in partial fulfilment of the requirements for the award of degree of* ***Bachelor of Engineering*** *in* ***Electronics & Communication Engineering*** *of* ***Visvesvaraya Technological University, Belagavi*** *during the academic year 2023-2024. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project has been approved as its satisfies the academic requirements in respect of project work prescribed for the bachelor of engineering degree.*

Project Guide

Dr. Hameed Miyan
Dept of ECE

Project Coordinator

Dr. Prashant Sangulagi
Dept of ECE

HOD

Dr. Prashant Sangulagi
Prof, Dept of ECE

Principal

Dr. Udaykumar Kalyane
BKIT, Bhalki

Name of the Examiners **Signature with Date**

1. _____ _____

2. _____ _____

S. V. E. Society's

Bheemanna Khandre Institute of Technology, Bhalki-585328

Department of Electronics & Communication Engineering

DECLARATION

We **Eshwar (3RB20EC010), Mahadev (3RB20EC017), Mouneshwar (3RB20EC019) & Prasad (3RB20EC022)** Students of 8th semester Bachelor of Engineering in Department of Electronics & Communication Engineering, **Bheemanna Khandre Institute of Technology, Bhalki** declare that the Project entitled **"NODE MCU BASED AIR AND NOISE MONITORING SYSTEM OVER IoT"** has been carried out by us and submitted in partial fulfilment of the requirements for the award of degree of **Bachelor of Engineering** in **Electronics & Communication Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year **2023-2024**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree.

Yours Sincerely,

ESHWAR (3RB20EC010)
MAHADEV (3RB20EC017)
MOUNESHWAR (3RB20EC019)
PRASAD (3RB20EC022)

Acknowledgements

First and foremost, we are grateful to our guide **Dr. Hameed Miyan** and Co-ordinator **Dr. Prashant Sangulagi** for his positive criticism, valuable suggestions and inevitable support. We felt extremely privileged and experienced an enthusiastic interest from his side. This fuelled our enthusiasm to step in even further and encouraged us a lot.

We take this opportunity to express our sincere thanks to **Dr. Prashant Sangulagi, HOD** and whole **Electronics & Communication Engineering** department for the constant encouragement during the completion of the presentation.

We would like to thank the principal **Dr. Udaykumar Kalyane**, BKIT Bhalki, for providing such a congenial working environment.

Last but not the least; we extend our gratitude to the Almighty for giving us strength and wisdom in completing this report.

Yours Sincerely,

ESHWAR (3RB20EC010)
MAHADEV (3RB20EC017)
MOUNESHWAR (3RB20EC019)
PRASAD (3RB20EC022)

ABSTRACT

In recent day scenario, the non-stop increase in air and sound pollution prove to be an big alarming problem. It has become mandatory to control and appropriately monitor the situation so that the required steps to control the situation can be undertaken. In this project, an IOT-based method using Nodemcu is used to monitor and check live the Air Quality Index and the sound pollution of a region, have been proposed. The recommended technology comprises of two modules namely, the Air Quality Index Monitoring Module, the Sound Intensity Detection Module. Firstly, the Air Quality Index is measured considering the presence of the air pollutants. Then the sound intensity is detected using respective sensor. System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to microcontroller. Also system keeps measuring sound level and reports it to the online server over IOT. The sensors interact with microcontroller which processes this data and transmits it over internet. This allows authorities to monitor air pollution in different areas and take action against it. Also authorities can keep a watch on the noise pollution near schools, hospitals and no honking areas, and if system detects air quality and noise issues it alerts authorities so they can take measures to control the issue.

Air and sound pollution is growing issue these days. It is necessary to monitor air quality for a better future and healthy living for all. We propose an air quality as well as sound pollution monitoring system that allows us to monitor and check air quality as well as sound pollution in particular area through IoT.

Air pollution is both an environmental and a social problem, as it leads to a multitude of adverse effects on human health, ecosystems and the climate. Quality of the air in city and urban areas is the most important factor that directly influences the incidence of diseases and decreases the quality of life. Appropriate soil water level is a necessary pre-requisite for optimum plant growth. Also, water being an essential element for life sustenance, there is the necessity to avoid its undue usage. Irrigation is a dominant consumer of water.

INDEX

Chapter 1

INTRODUCTION

The system proposed in this is an advanced solution for monitoring the environmental conditions at a particular place and make the information visible anywhere in the world. The technology behind this is Internet of Things (IoT), which is an advanced and efficient solution for connecting the things to the internet and to connect the entire world of things in a network. Here things might be whatever like electronic gadgets, sensors and automotive electronic equipment. The system deals with monitoring and detection of air and sound pollution and sends sensor data to the computer and android phone using IoT link. The data updated from the implemented system can be accessible in the internet from anywhere in the world using think speak plate form.

In recent day scenario, the non-stop increase in air and sound pollution proves to be a big alarming problem. It has become mandatory to control and appropriately monitor the situation so that the required steps to control the situation can be undertaken. In this project, an IoT-based method using Node MCU is used to monitor and check live the Air Quality Index and the sound pollution of a region, have been proposed. The recommended technology comprises of two modules namely, the Air Quality Index Monitoring Module, the Sound Intensity Detection Module. Firstly, the Air Quality Index is measured considering the presence of the air pollutants. Then the sound intensity is detected using respective sensor. System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to microcontroller. Also, system keeps measuring sound level and reports it to the online server over IoT.

The sensors interact with microcontroller which processes this data and transmits it over internet. This allows authorities to monitor air pollution in different areas and take action against it. Also, authorities can keep a watch on the noise pollution near schools, hospitals and no honking areas, and if system detects air quality and noise issues it alerts authorities so they can take measures to control the issue.

1.1 PROBLEM DEFINITION

Objective of this project is to design and develop IoT based air and sound pollution. The system consists of air quality sensor and sound sensor. Sensors sends the data to computer using IoT link. User can monitor on think speak. The system will alert if any sensors reading goes

above the pre-set level.

An effective natural observing framework is essential to screen and estimate the conditions in the event of surpassing endorsed level of parameter (for example Carbon Mono-oxide (CO) and radiation levels). At the point when the items like condition furnished with sensor gadgets, smaller scale controller and different programming application turn into a self-securing and self-observing condition.

1.2 OBJECTIVES

1. To study the existing system.
2. To design the block diagram.
3. To decide the components specification & device in system.
4. To design the circuit diagram and simulate it using suitable software.
5. To design the PCB and implement hardware.
6. To test the circuit and observe the result.
7. To prepare report.

Chapter 2

LITERATURE REVIEW

A weather station is a facility, either on land or sea, with instruments and equipment for measuring atmospheric conditions to provide information for weather forecasts and to study the weather and climate. The measurements taken include temperature, atmospheric pressure, humidity, wind speed, wind direction, and precipitation amounts. Wind measurements are taken with as few other obstructions as possible, while temperature and humidity measurements are kept free from direct solar radiation, or insolation. Manual observations are taken at least once daily, while automated measurements are taken at least once an hour. Weather conditions out at sea are taken by ships and buoys, which measure slightly different meteorological quantities such as sea surface temperature (SST), wave height, and wave period. Drifting weather buoys out number their moored versions by a significant amount.

2.1 Poonam Pal1, Ritik Gupta2,Sanjana Tiwari3,Ashutosh Sharma4 **"IoT Based Air Pollution Monitoring System Using Arduino"** Oct -2017

The level of pollution has increased with times by lot of factors like the increase in population, increased vehicle use, industrialization and urbanization which results in harmful effects on human wellbeing by directly affecting health of population exposed to it. In order to monitor In this project we are going to make an IOT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a web server using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO2, smoke, alcohol, benzene and NH3. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily. In this IOT project, you can monitor the pollution level from anywhere using your computer or mobile. The main objective of IOT Air & Sound Monitoring System is that the Air and sound pollution is a

growing issue these days. It is necessary to monitor air quality and keep it under control for a better future and healthy living for all. Due to flexibility and low cost Internet of things (IoT) is getting popular day by day. With the urbanization and with the increase in the vehicles on road the atmospheric conditions have considerably affected. Harmful effects of pollution include mild allergic reactions such as irritation of the throat, eyes and nose as well as some serious problems like bronchitis, heart diseases, pneumonia, lung and aggravated asthma. Monitoring gives

measurements of air pollutant and sound pollution concentrations, which can then be analyzed interpreted and presented. This information can then be applicable in many ways. Analysis of monitoring data allows us to assess how bad air pollution and sound pollution is from day to day.

2.2 Ms. Sarika Deshmukh Mr. Saurabh Surendran Prof M.P. Sardey **"Air and Sound Pollution Monitoring System using IoT"**

As modernization is growing rapidly internet technologies and wireless sensor networks are advanced, a new trend in the era of omnipresence is being realized. The increase in the number of internet users and application on the internet working technologies enable networking of everyday objects requiring human-to-human or human-to-computer communication. Internet of Things allows an exchange of information to and from a device or thing. It can be anything such as refrigerators, watches, fans, air conditioner, automobiles, or anything. It is a communication between human and machine or machine and machine. Due to flexibility and low-cost Internet of things (IoT) is getting popular day by day. With the urbanization and with the increase in the vehicles on road the atmospheric conditions have considerably affected. Also, there has been the growth of industries and infrastructure which has caused increase in pollution in atmosphere like air and sound pollution. Air pollution and sound pollution are major constituents for having adverse and harmful effects on environment as well on human beings. To monitor this pollution is a very difficult task. Traditionally, authorities like data loggers were used to collect the data of the site to be analyzed. They had to visit the site to be analyzed every time they wanted the data. This was a lengthy, time consuming and expensive task. Due the use of sensors collaborated with internet can make pollution monitoring less complex, less time consuming and flexible. The data can be obtained from remote location without having to visit the location due the internet.

Chapter 3

METHODOLOGY AND RELATED WORK

3.1 Block Diagram

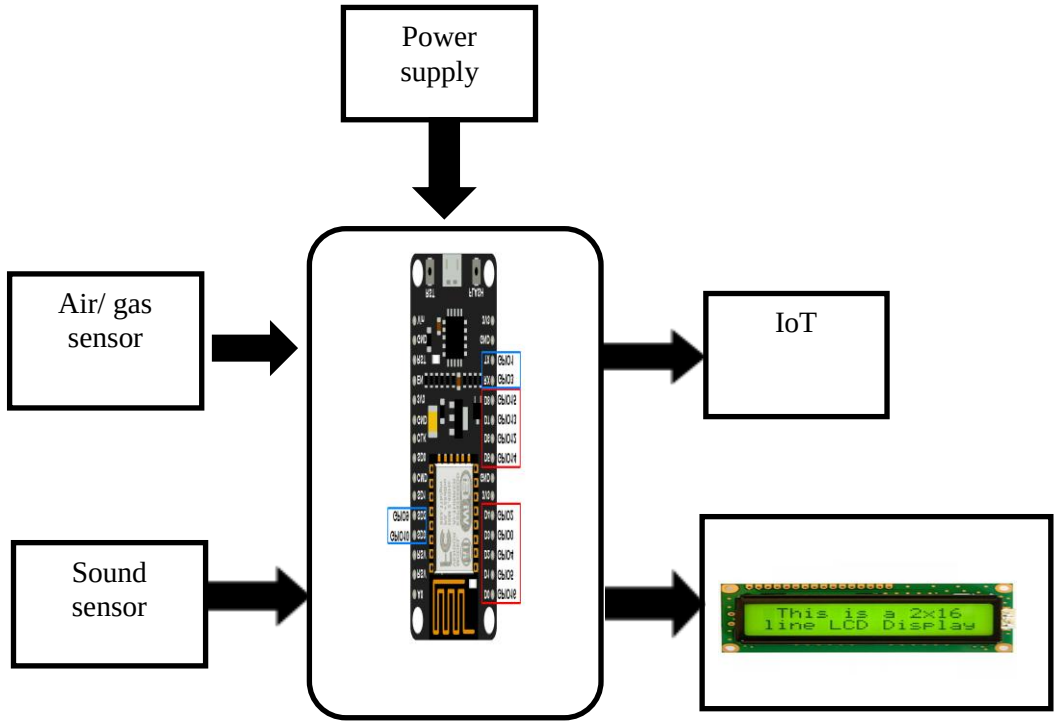


Fig 3.1: Block Diagram

As shown in fig. in system we use Arduino as main controller. In system we use MQ135 gas sensor for detecting or sensing gases and also use sound sensor LM393 module for detect the sound pollution. Sensed data of sensor given to analog pin of the Arduino then digital output pin are connected to LCD, buzzer and LED. If air pollution is there then buzzer will start beeping and if sound pollution is there then LED will glow. All condition of pollution display on LED

Working

- ✓ Air quality sensor for monitoring air quality.
- ✓ Condenser mic for sound monitoring.
- ✓ All the sensors are attached at the ADC pin (analog to digital) to the Arduino microcontroller.
- ✓ Microcontroller will do some calculation and send data to IoT module.

- ✓ We will set the limit of every sensor if limit exceeded it will alert the user.by displaying limit crossed.
- ✓ IoT module will transmit data so that computer will receive and display on screen.
- ✓ Computer will display data on think speak plate form.

3.2 Related work :

There are numerous works that have been done related to IOT based Projects.

Kavitha.B.C *et al.* [1] given an idea to realize the IoT system by introducing raspberry pi and IOT shield. They use 3 different gas sensors, namely mq7, mq135 and DHT 11. The air quality can be displayed and monitored on LCD simultaneously available on webpage.

Ajitesh Kumar, Mona Kumari, Harsh Gupta, GLA University Mathura, India.[2] They used nodemcu to monitoring air and noise which present in atmosphere MQ 2 gas sensor used to sense the smoke and MQ9 gas sensor used to sense carbon monoxide . PMS3003 G3 particle they used pm2.5 giving reading to node mcu processor and data to the rends. The dada internet. They also used a tod converter a dc to convert and leg data. To digital. They used OLED display, to display the pollution condition. All sensors will sends data to node mcu and and they node mcu send to think speak then graph shows on think speak and also data show on a LED.

V. Karthika1 S. Nandhini2, P. Preethy3 S. Sivapriya4, M.Thangabrindha5[3] In this system used PIC 16F877 Module for the Industrial Air monitoring system. The remote monitoring & controlling of Air quality of the room inside a building can designed the main purpose of constructing the proposed system.

Chapter 4

REQUIREMENT

Requirements specification refers to specific design requirement. It's the process of writing down the user and system requirements into a document. The requirements should be clear, easy to understand, complete and consistent.

In this chapter we discussed the various components that are used to design the proposed model.

4.1 Hardware Components

Hardware requirements. The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems.

1) Esp32 :

Fig 4.1 ESP8266

The ESP8266 Node MCU CP2102 board has ESP8266 which is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

Features:

• Small volume, easily embedded to other products

• Strong function with support LWIP Protocol, Freertos

• Supporting three modes : AP, STA, and AP+STA

• Supporting Lua program, easily to develop Includes CP2102 USBUART bridge

Note: The ESP8266 Module is not capable of 5-3V logic shifting and will require an external Logic Level Converter. Please do not power it directly from your 5V dev board.

Note: This new version of the ESP8266 Wi-Fi Module has increased the flash disk size from 512k to 1MB.

ESP32 Peripherals

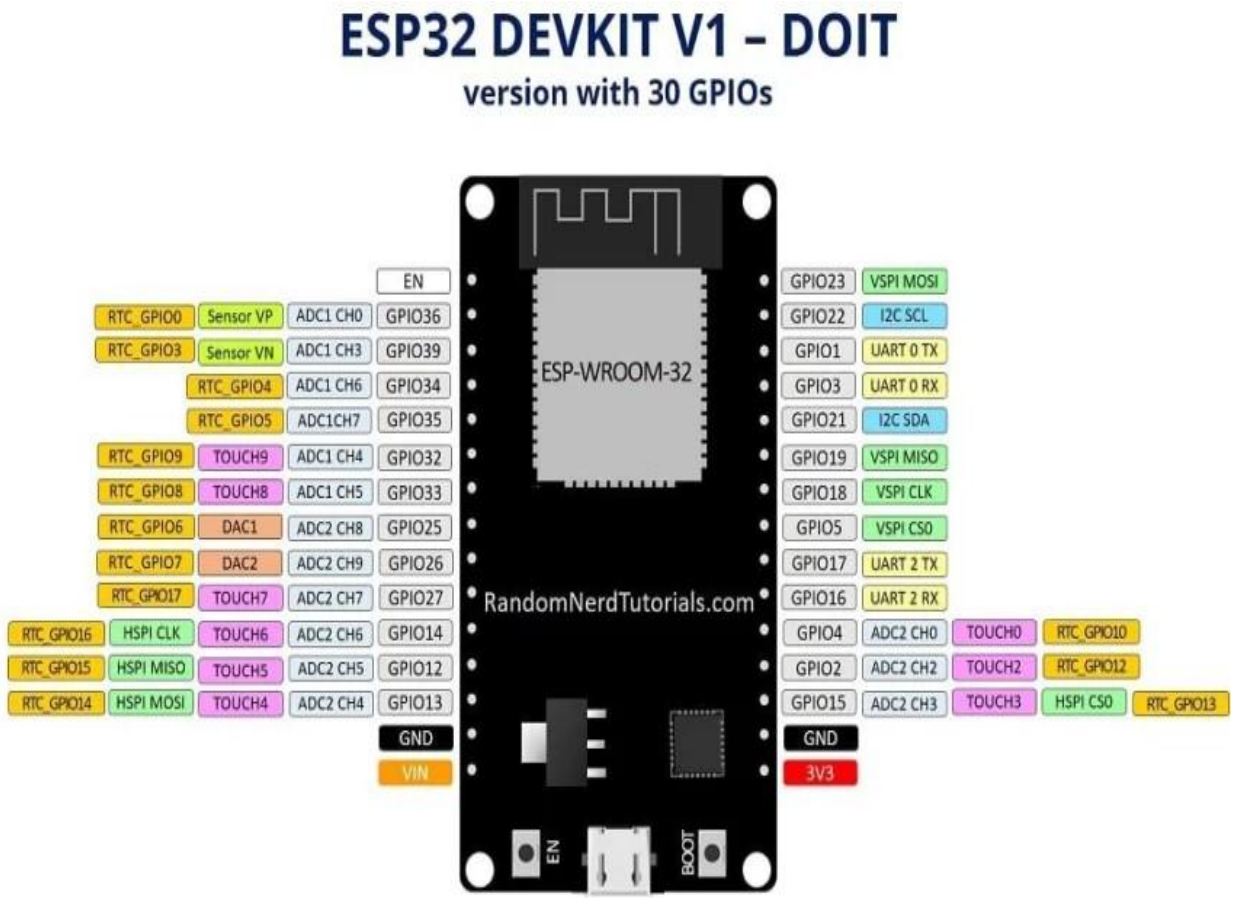


Fig 4.2 ESP32 Peripherals

• 18 Analog-to-Digital Converter (ADC) channels

• 3 SPI interfaces • 3 UART interfaces

• 2 I2C interfaces • 16 PWM output channels

• 2 Digital-to-Analog Converters (DAC)

• 2 I2S interfaces

• 10 Capacitive sensing GPIOs

The ADC (analog to digital converter) and DAC (digital to analog converter) features are assigned to specific static pins. However, you can decide which pins are UART, I2C, SPI, PWM etc . We just need to assign them in the code. This is possible due to the ESP32 chips multiplexing feature.

There are totally 39 digital Pins on the ESP32 out of which 34 can be used as GPIO and the remaining are input only pins. The device supports 18channels for 12-bit ADC and 2-channel

for 8-bit DAC. It also has 16 channels for PWM signal generation and 10 GPIO pins supports capacitive touch features.

MQ135 Gas Sensor:

Fig 4.3 gas sensor

The MQ135 is a gas sensor it used for detecting or sensing harmful gases in the atmosphere. It has wide detecting scope. It gives fast response and also it high sensitivity sensor. It is simple and long-life device. They are used in air quality control equipment for building offices are suitable for detecting of NH3, alcohol, benzene, smoke CO2 etc.

Feature

- Wide detecting scope
- Fast response and High sensitivity
- Stable and long life • Operating Voltage is +5V
- Detect/Measure NH3, NOx, alcohol, Benzene, smoke, CO2, etc.
- Analog output voltage: 0V to 5V

3 LM393 Sound Sensor:

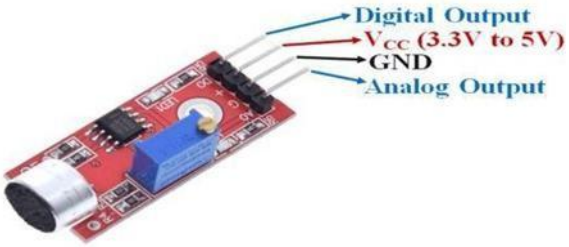


Fig 4.4 Sound sensor

The sound sensor module provides an easy way to detect sound and it generally used for detecting sound intensity. Module detect the sound has exceeded a threshold value. Sound is detected via microphone and fed into an LM393 op-amp. The sound level adjust through pot. The sound increases set value then output is low. These module work on DC 3.3-5 voltage.

Feature

- Operating voltage 3.3V-5V
- Output model: digital switch outputs (0 and 1, high or low level)
- Voltage Gain 26dB
- Microphone Impedance 2.2kΩ
- Microphone Frequency 16.20 kHz

Advantages:

They are used for security systems.

- ✓ Often it works with speech recognition software where in sound or speech is converted to text. This is faster approach compare to typing using keyboard. This is also used by disabled people.
- ✓ It is easy to manipulate sound in real time. Hence earlier recording is not needed.
- ✓ The sound sensors are used while driving by switching on the radio. This helps in improving safety using voice activation system based on sound sensor.
- ✓ They are easily affordable.
- ✓ Wireless sound sensors provide great freedom while speaking. Moreover, it does not require cabling compare to wired mic.

4 16*2 LCD Display:

LCD is used for to display the condition there are three conditions in air pollution and three conditions in noise pollution means air and sound is clear, moderately polluted or highly polluted that is displayed on LED.

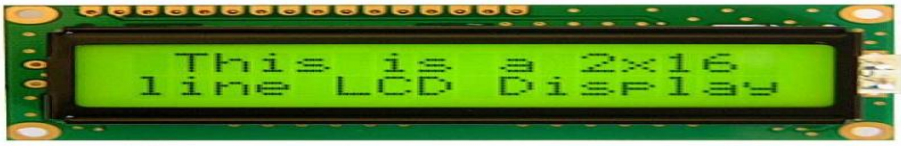


Fig 4.5 LCD display

Feature

- Operating Voltage is 4.7V to 5.3V
- Current consumption is 1mA without backlight
- Alphanumeric LCD display module, meaning can display alphabets and numbers
- Consists of two rows and each row can print 16 characters.

4.2 Software requirement

A software requirements specification(SRS) is a description of a software system to be developed. The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide

4.2.1 C++

"C++ is a statically-typed, free-form, (usually) compiled, multi-paradigm, intermediate-level general-purpose middle-level programming language." In simple terms, C++ is a sophisticated, efficient and a general-purpose programming language based on C. It was developed by Bjarne Stroustrup in 1979.Many of today's operating systems, system drivers, browsers and games use C++ as their core language. This makes C++ one of the most popular languages today. A language is said to be platform dependent whenever the program is execute in the same operating system where that was developed and compiled but not run and execute on other operating system. C++ is platform dependent language.

Hardware – Recommended

Microsoft* Windows* Systems

- ✓ Intel(R) Core(TM) processor family
- ✓ Intel(R) Xeon(R) processor family
- ✓ Intel(R) Xeon Phi(TM) processor family

Linux* Systems

- ✓ Intel(R) Core(TM) processor family
- ✓ Intel(R) Xeon(R) processor family
- ✓ Intel(R) Xeon Phi(TM) processor family

macOS* Systems

- ✓ Intel(R) Core(TM) processor family

Android* Systems

- ✓ Intel(R) Atom(TM) processor family Software - Minimum Requirements
- ✓ Supported operating system (see below)

Supported compiler (see below)

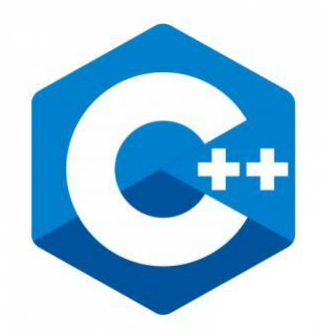


Fig 4.6 C++ Language

4.2.2 Arduino IDE programming microcontroller:

Fig 4.7 Arduino IDE

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to notices and experts like.

Chapter 5

DESIGN AND SYSTEM ANALYSIS

Design is the creation of a plan or convention for the construction of an object, system or measurable human interaction (as in architectural blueprints, engineering drawings, business processes, circuit diagrams, and sewing patterns). In some cases, the direct construction of an object (as in engineering, management, coding, and graphic design) is also considered to use design thinking.

The project analysis provides critical data support to a technical team. Analysis functions may include budget tracking and financial forecasting, project evaluation and monitoring, maintaining compliance with corporate and public regulations, and performing any data analysis relevant to project tasks.

5.1 Modules of projects

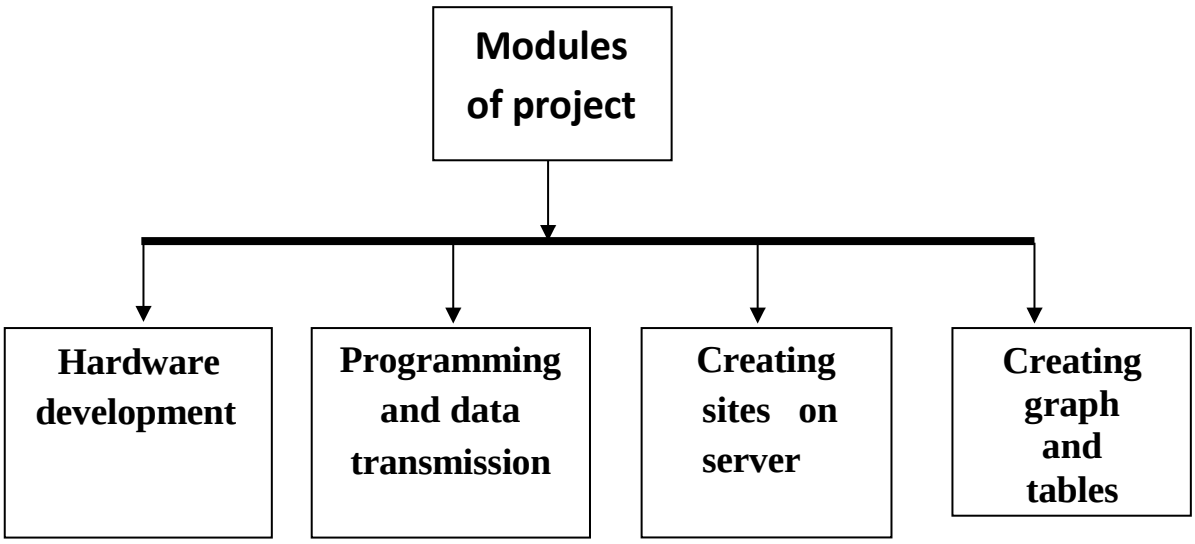


Fig 5.1: Modules of projects

5.2 Data flow diagram:

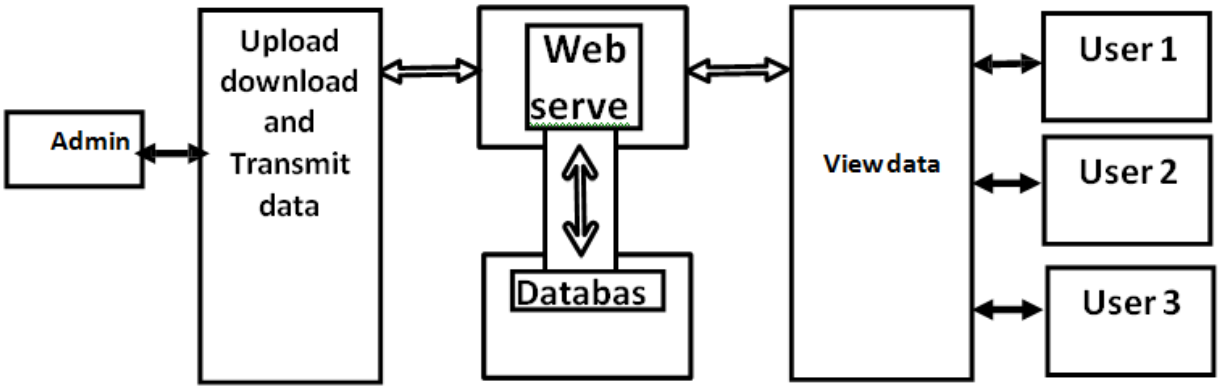
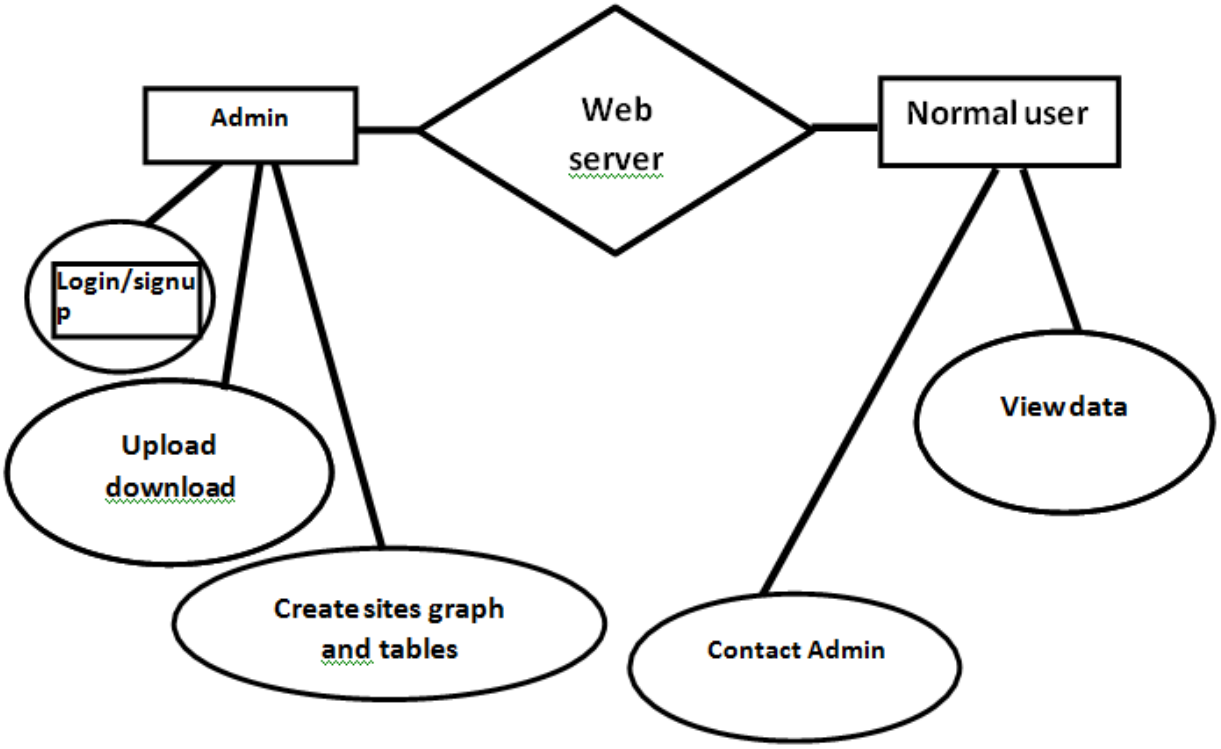


Fig 5.2: DFD

5.3 ER Diagram:



5.3 Action Plan:

S. no	Description	Date
1	Circuit designing and component selection	30Sep 23
2	PCB designs and components mounting	30oct 23
3	Coding of firmware	15Jan 24
4	Integration of all module	15Mar 24
5	Testing and Documentation	28Mar 24

5.4 Project modules:

- ✓ Module 1: Circuit designing and component selection.
- ✓ Module 2: PCB designing and components mounting.
- ✓ Module 3: Coding of firmware.
- ✓ Module 4: Integration of all module and testing
- ✓ Module 5: Documentation.

Chapter 6

TESTING

Testing- Testing is a key step. Testing can be defined as checking for correct behavior prior to release to end users. It is process of testing software for defects

AVR programming

In this post I will walk you through the steps required to program an AVR chip using the AVRISP mkII.

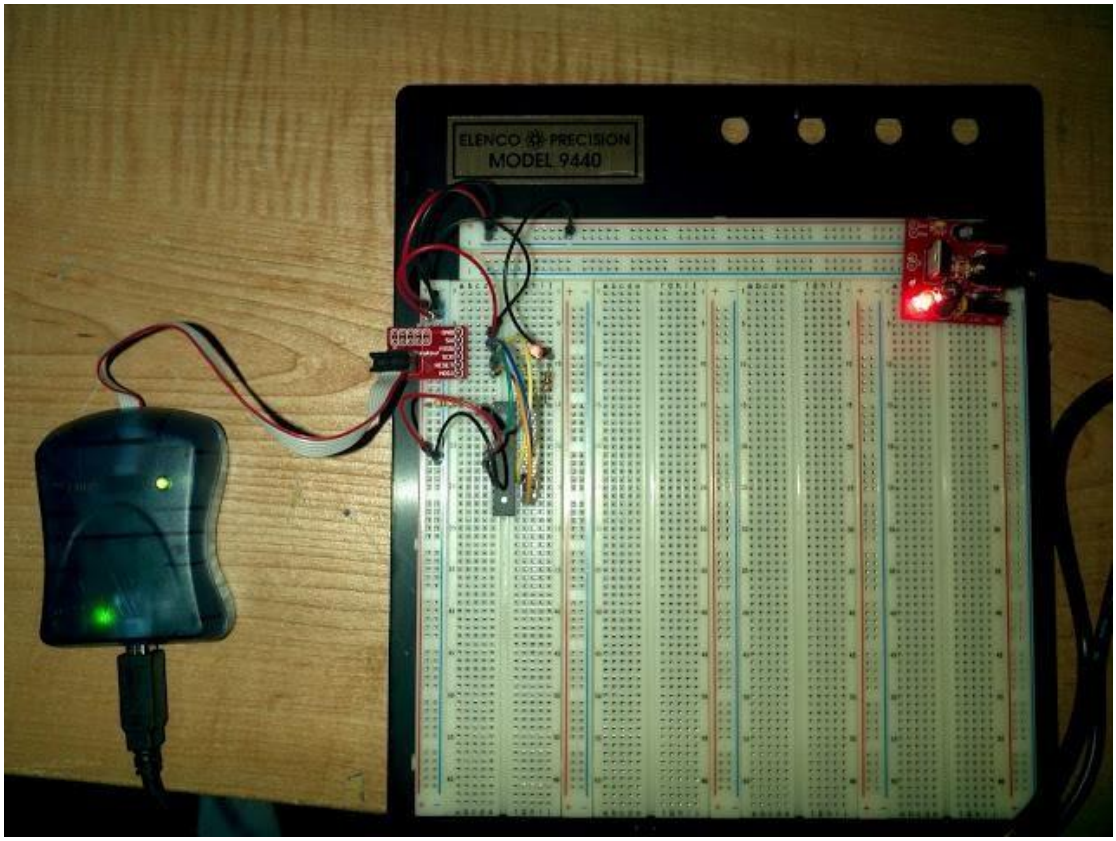


Fig6.1: AVR chip, programmer, adapter, breadboard and power supply

Hardware and Software Required: Here is a list of things you'll need **Hardware:**

- ✓ 4.7K resistor
- ✓ 1k resistor
- ✓ LED
- ✓ AVR Chip: I'll use the atmega328p

- ✓ Programmer: AVRISP mkII
- ✓ AVR Programmer Adapter (optional)
- ✓ Jumper Wires
- ✓ 5V Power supply. You can of course use batteries if you prefer.

Software:

- ✓ AVR Studio 5

Circuit Set Up:

Simply find the pins in your programmer, match them to the pins in your chip and connect them. Also connect the resistor from pin 7 to power.

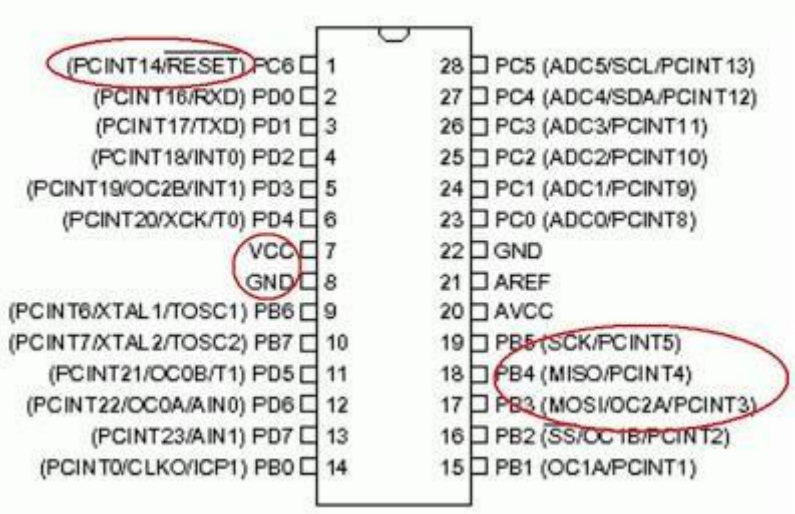
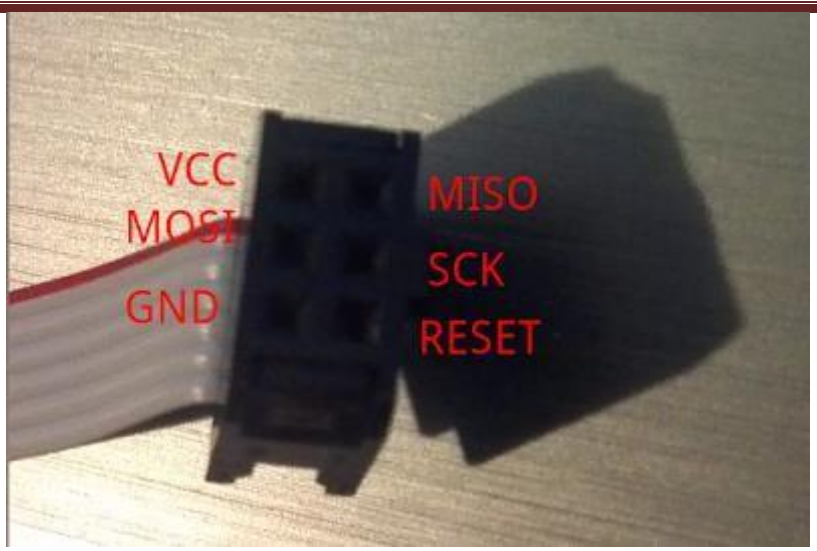


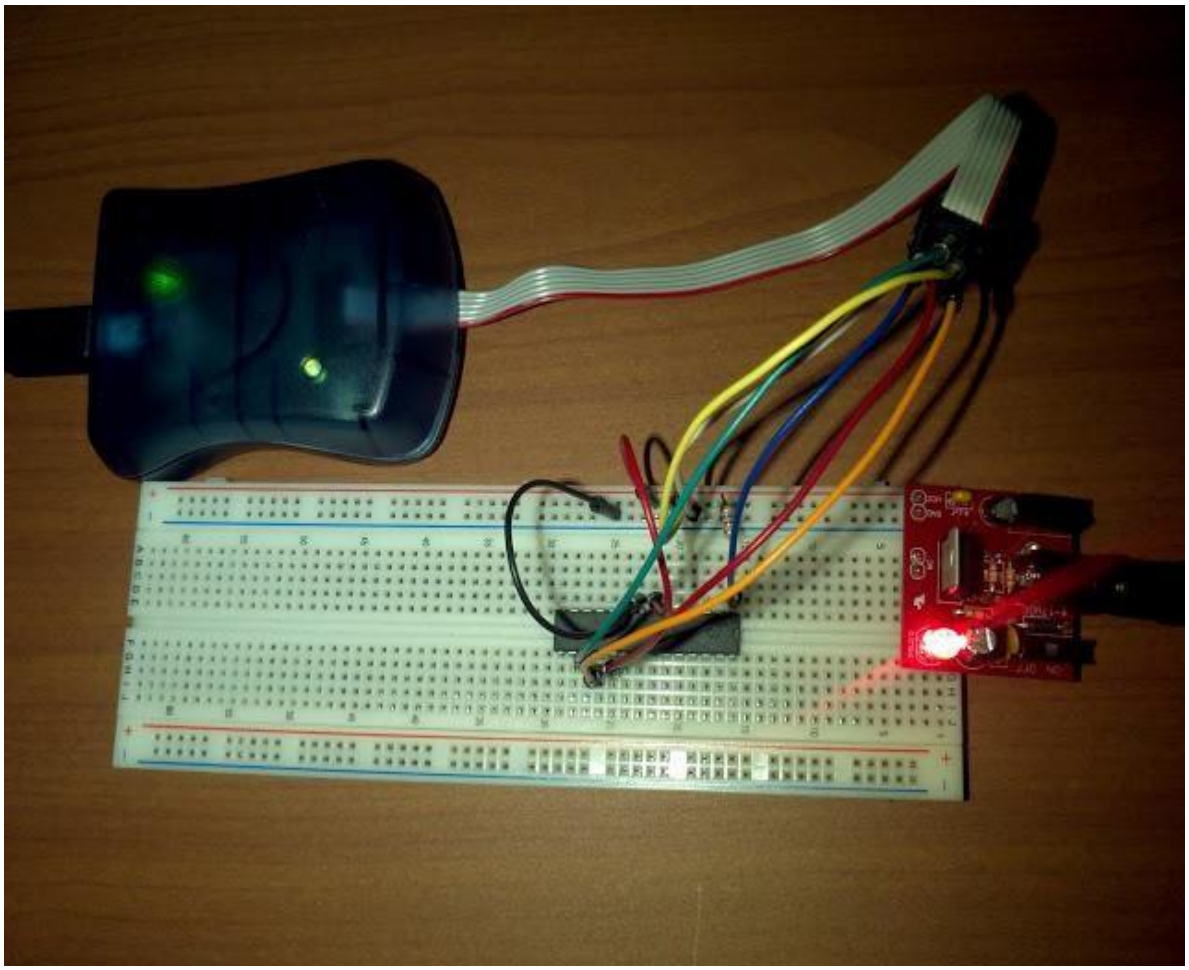
Fig 6.2 ATmega328p pin out

Now because the programmer doesn't have one squared hole, or at least mine doesn't, it has many, let me show how how that looks in real life, and yes it is like a mirror image of the image above (weird right?).



AVR is pmk II holes

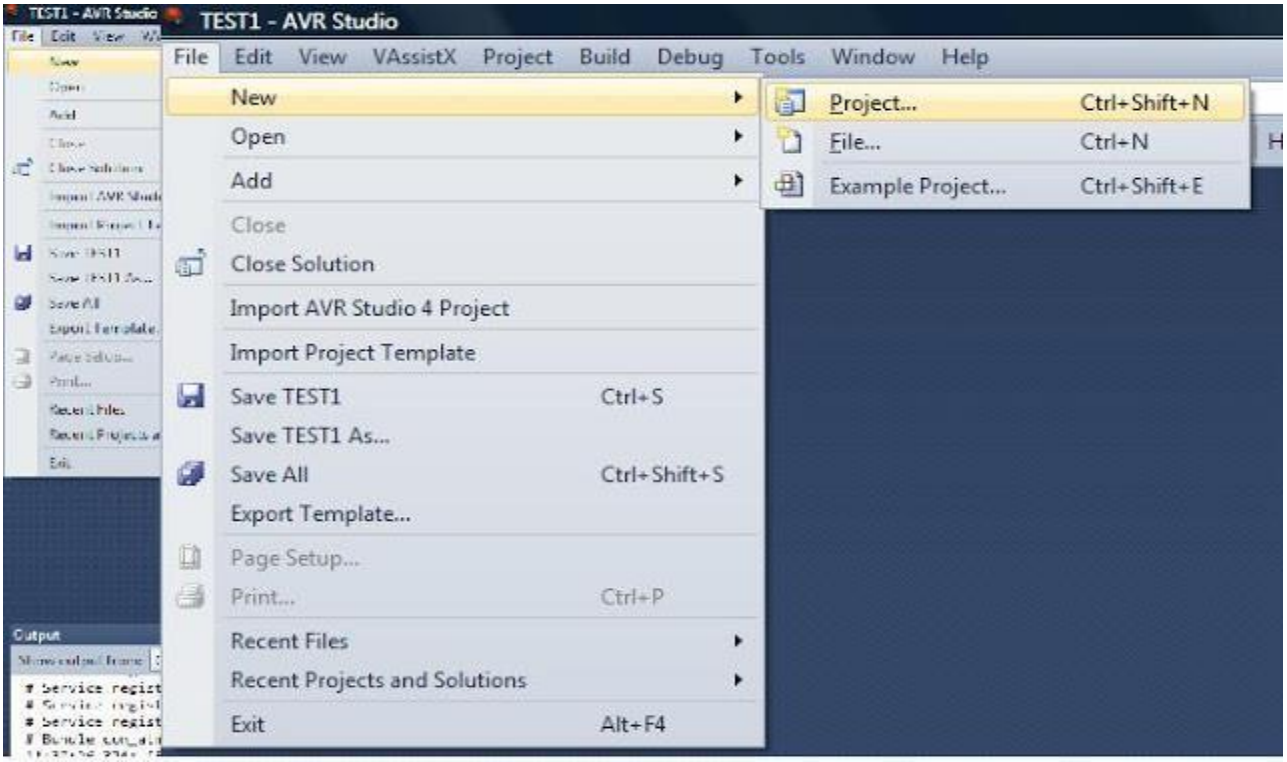
Now that we have everything connected…



AVRISP mk II: circuit to program ..let's move on to the software. Check out the picture at the very top of this page to see how much neater your circuit will be with the programming adapter.

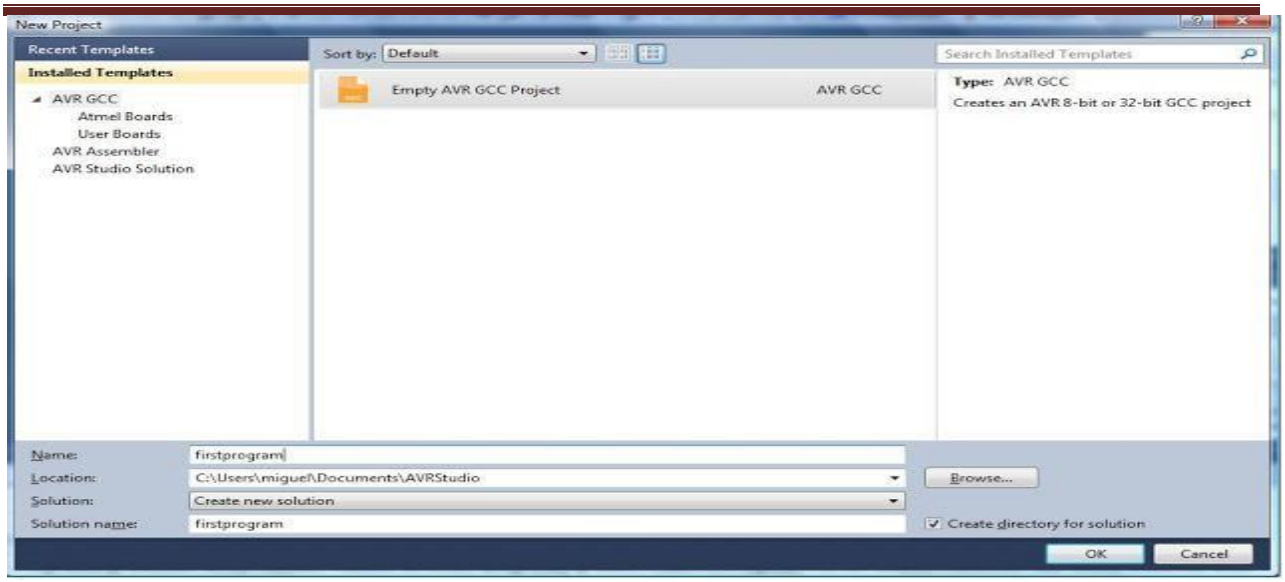
Our First AVR Program

Let me go ahead and walk you through the steps of using AVR Studio 5. Once you have opened AVR Studio click on File ->New -> Project.



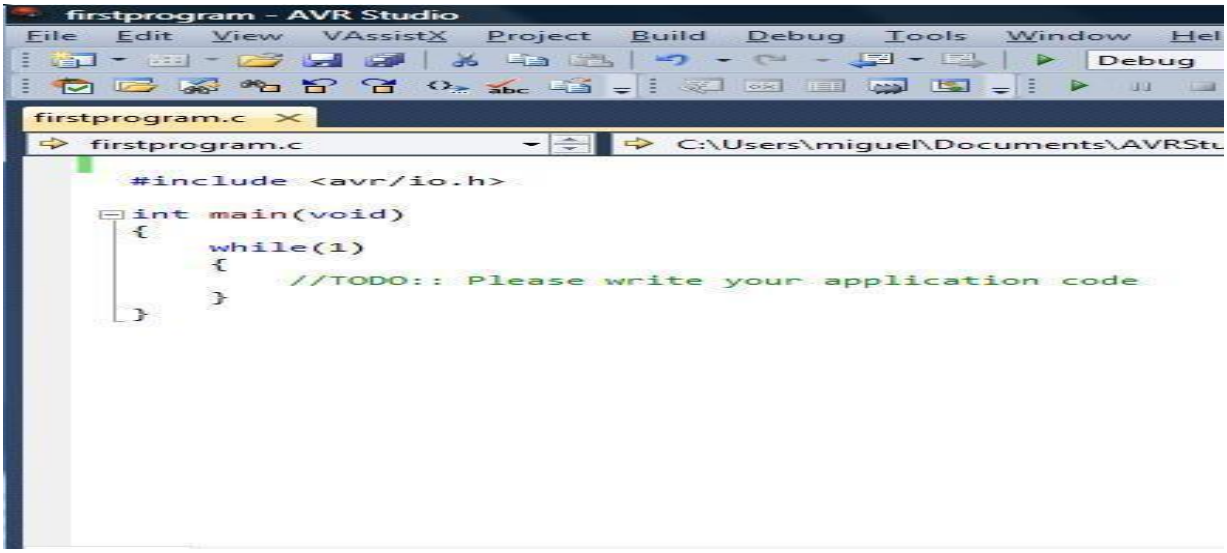
Creating a new project in AVR Studio 5

Name your project, I named mine first program. Now click the OK button.



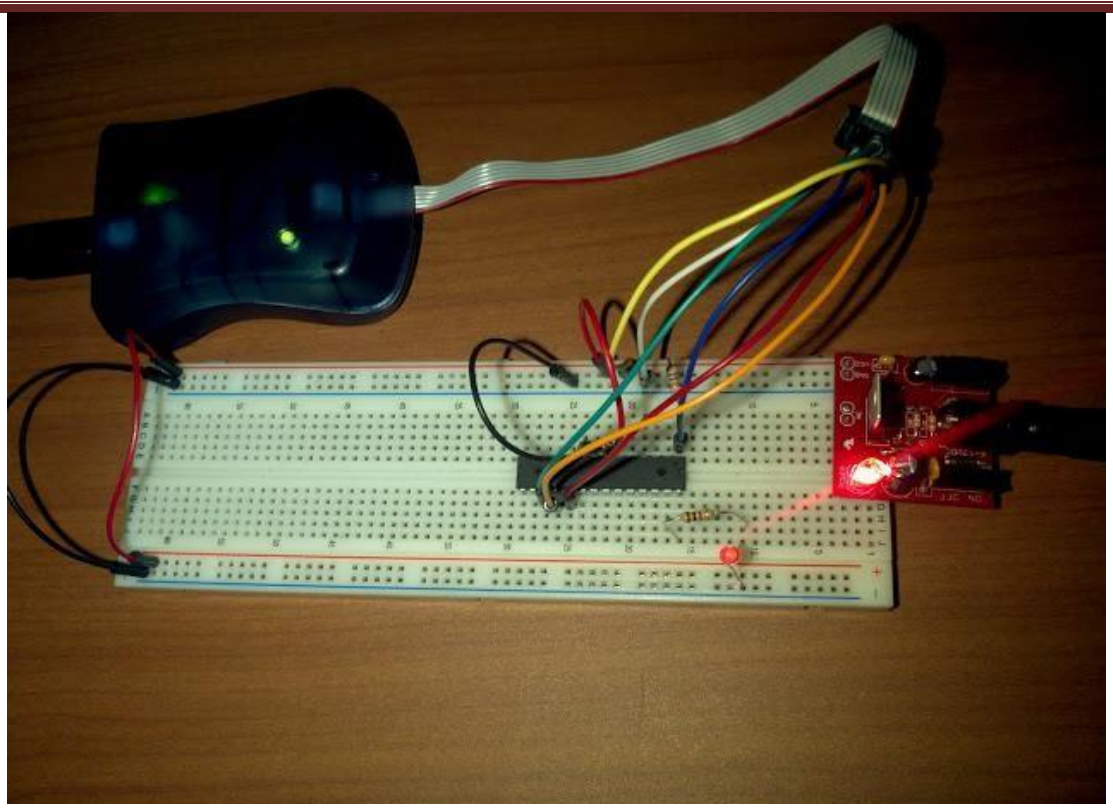
Naming your project

Next you will be greeted with an almost blank C file.



AVR Studio blank template C file

So what software are we going to write? how about we make an led blinker. Connect an LED to port C5 of your microcontroller through the 1k resistor and then to ground.



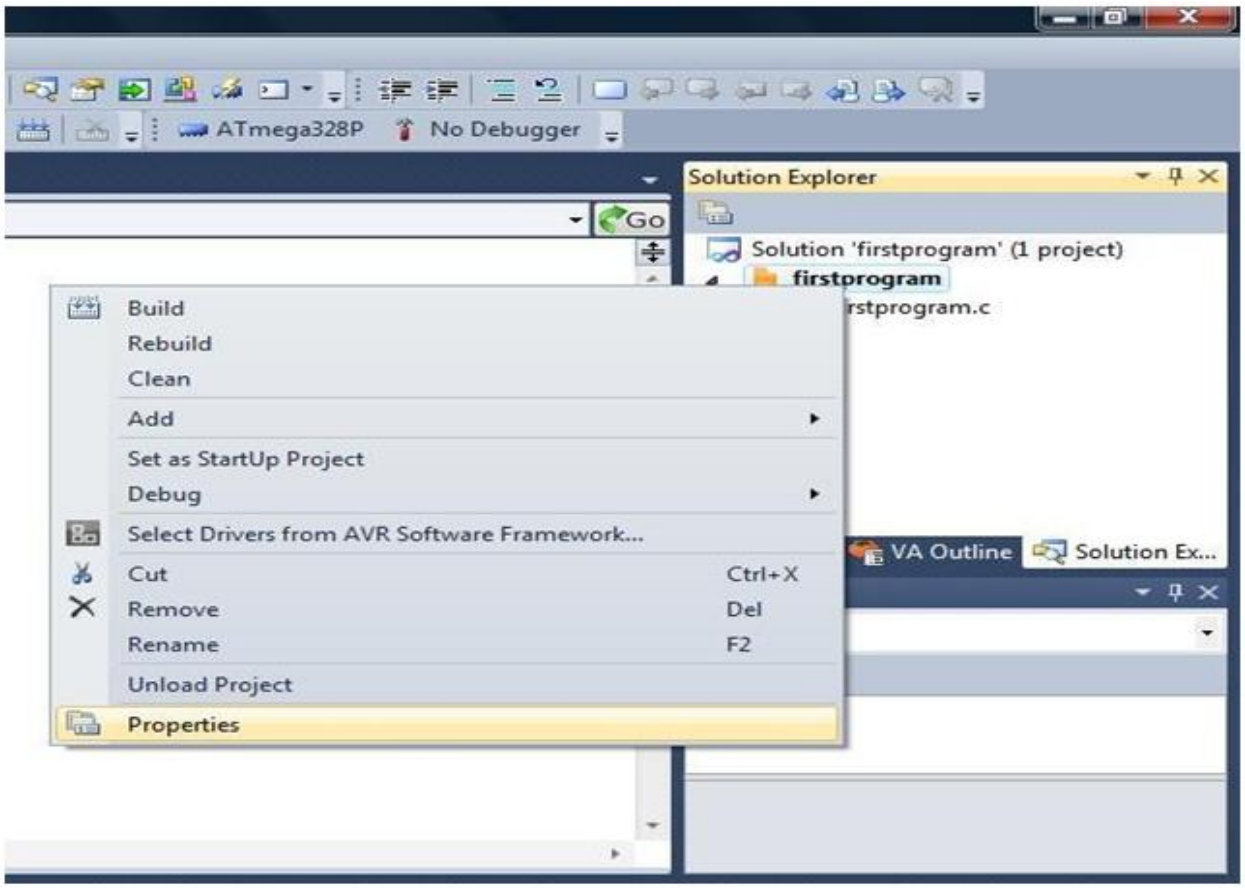
red led connected to port C5 (pin 28) of the ATmega328p

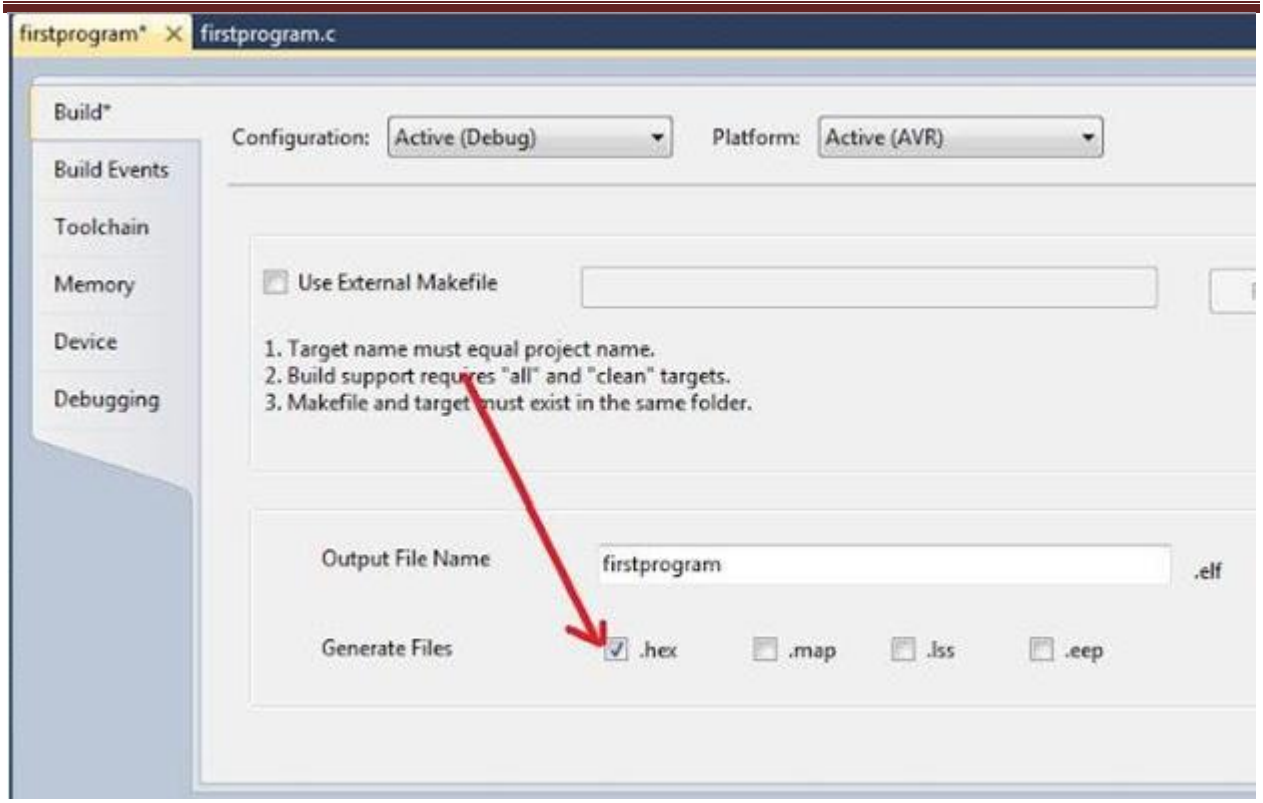
Now back to AVR Studio. Replace the text in the file with the following.

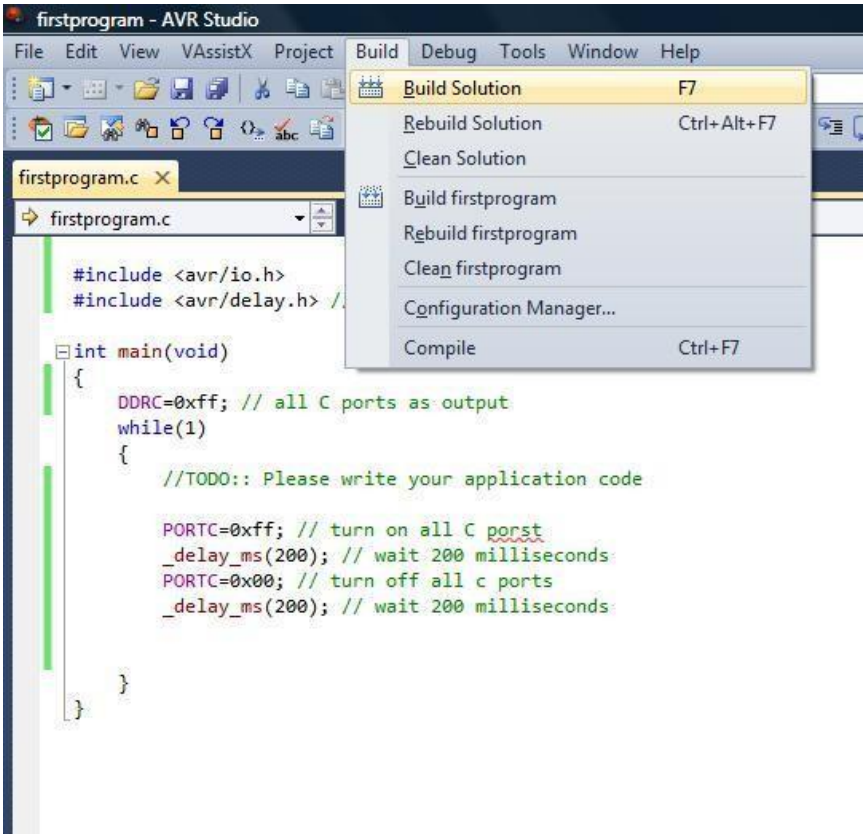
```
#include<avr/io.h>

#include<avr/delay.h>// has some delay functions you can use 3.

int main(void)

{

DDRC=0xff;// all C ports as output

while(1)

{

//TODO:: Please write your application code 10.

PORTC=0xff;// turn on all C porst

_delay_ms(200);// wait 200 milliseconds

PORTC=0x00;// turn off all c ports

_delay_ms(200);// wait 200 milliseconds

}
```

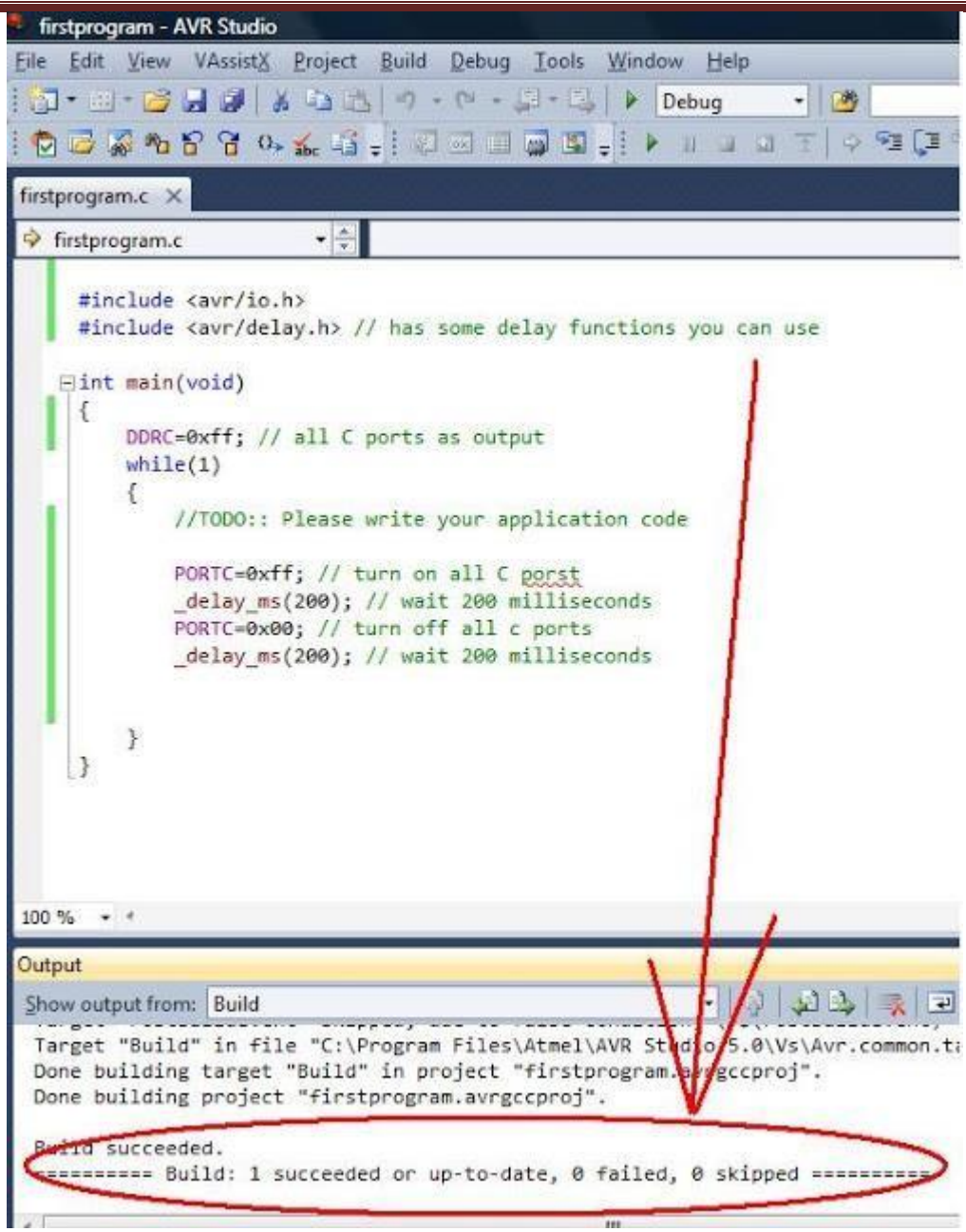
}

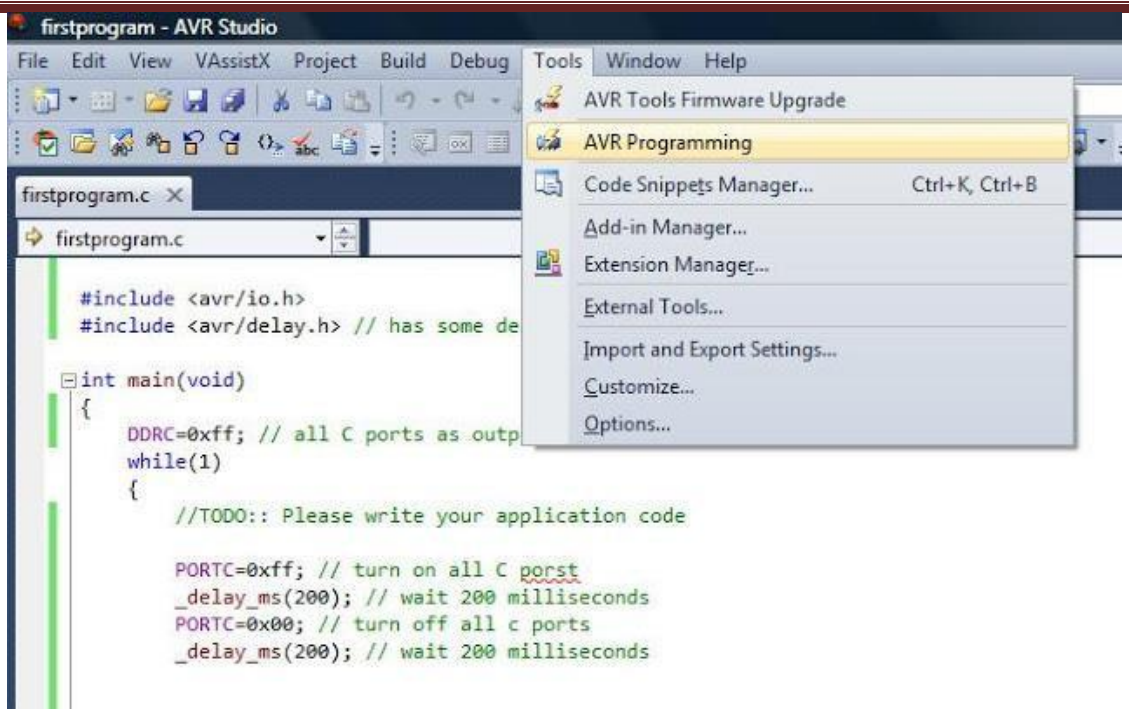
The code that gets uploaded to the microcontroller is actually a HEX file, but we have to tell AVR Studio to make this file. To do that right click on your project's name on the right in the solution explorer window and click on The code that gets uploaded to the microcontroller is actually a HEX file, but we have to tell AVR Studio to make this file. To do that right click on your project's name on the right in the solution explorer window and click on properties.

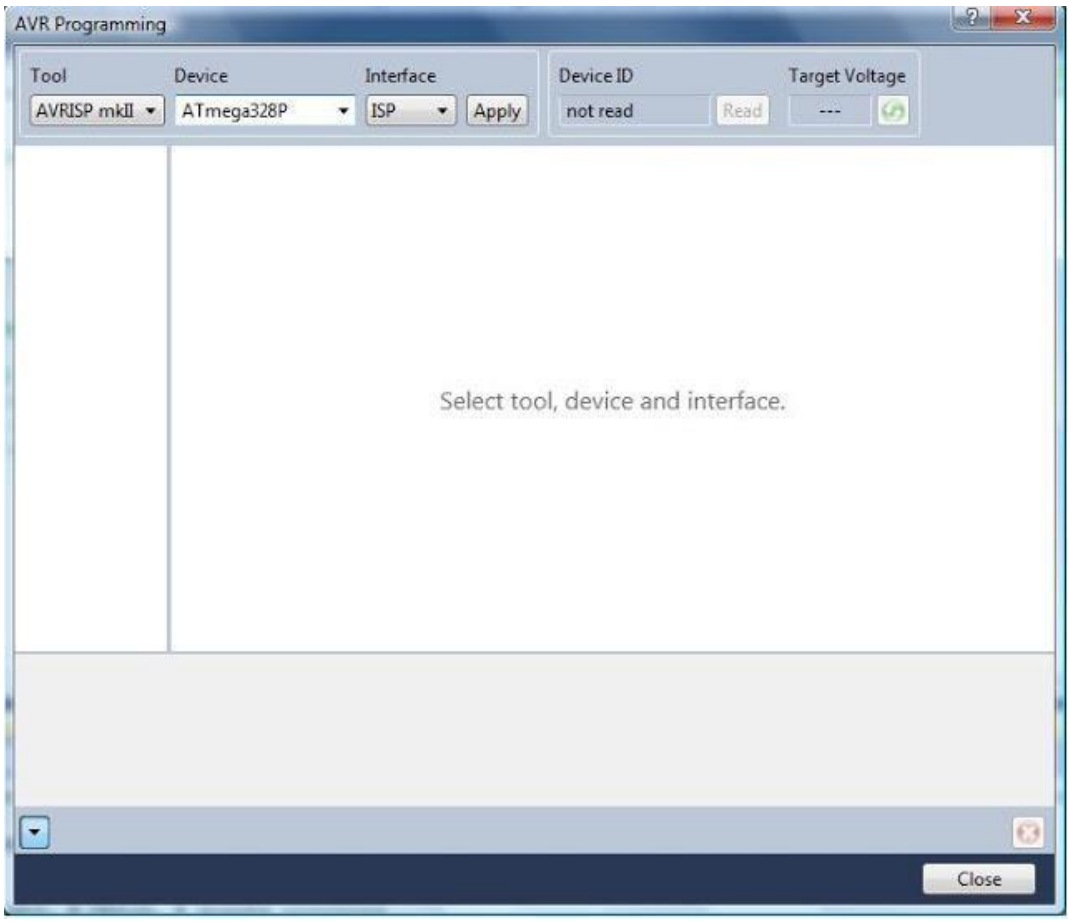



firstprogram*
firstprogram.c
Build*
Build Events
Toolchain
Memory
Device
Debugging
Configuration: Active (Debug)
Platform: Active (AVR)
Use External Makefile
1. Target name must equal project name.
2. Build support requires "all" and "clean" targets.
3. Makefile and target must exist in the same folder.
Output File Name
firstprogram
.elf
Generate Files
.hex
.map
.lss
.eep

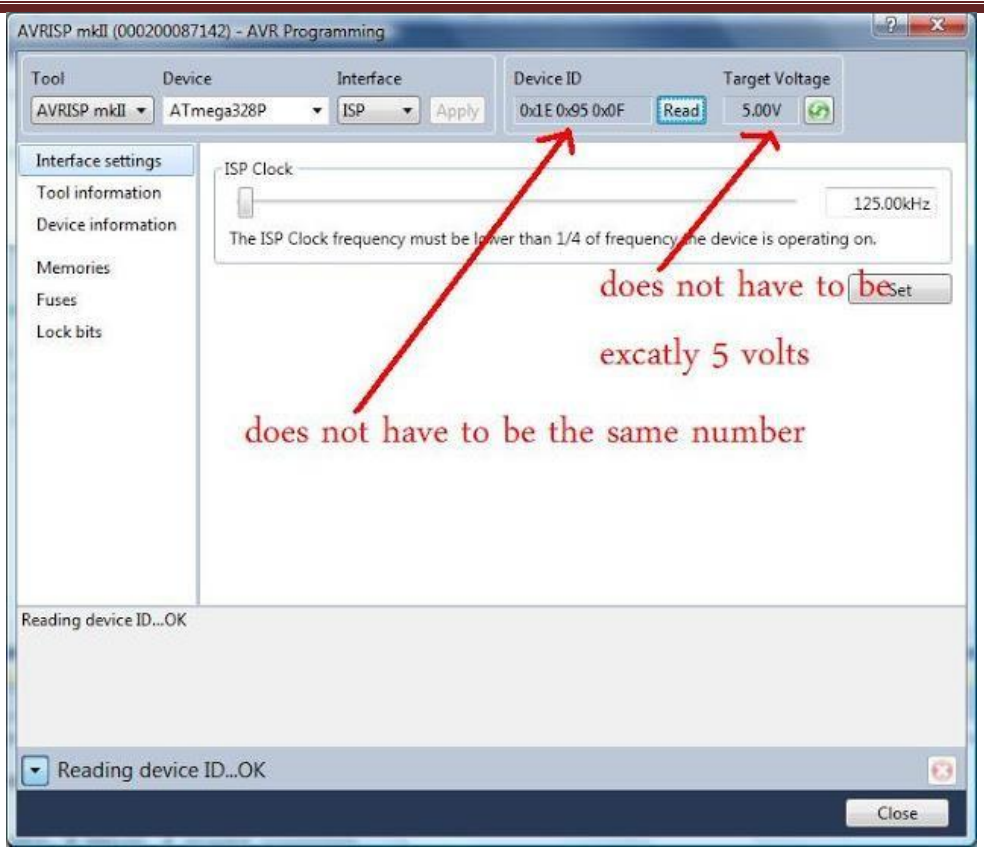

firstprogram - AVR Studio
File Edit View VAssistX Project Build Debug Tools Window Help
Build Solution F7
Rebuild Solution Ctrl+Alt+F7
Clean Solution
Build firstprogram
Rebuild firstprogram
Clean firstprogram
Configuration Manager...
Compile Ctrl+F7
firstprogram.c
#include <avr/io.h>
#include <avr/delay.h>
int main(void)
{
DDRC=0xff; // all C ports as output
while(1)
{
//TODO:: Please write your application code
PORTC=0xff; // turn on all C porst
_delay_ms(200); // wait 200 milliseconds
PORTC=0x00; // turn off all c ports
_delay_ms(200); // wait 200 milliseconds
}
}

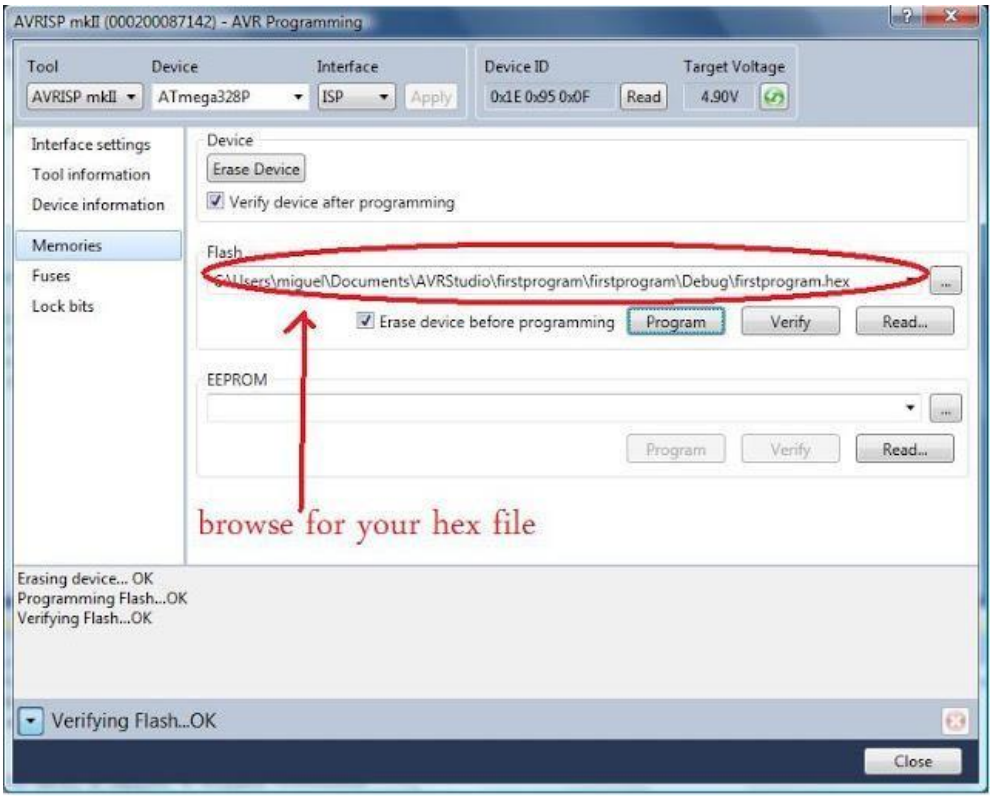

firstprogram - AVR Studio
File Edit View VAssistX Project Build Debug Tools Window Help
Debug
firstprogram.c
firstprogram.c
#include <avr/io.h>
#include <avr/delay.h> // has some delay functions you can use
int main(void)
{
DDRC=0xff; // all C ports as output
while(1)
{
//TODO:: Please write your application code
PORTC=0xff; // turn on all C porst
_delay_ms(200); // wait 200 milliseconds
PORTC=0x00; // turn off all c ports
_delay_ms(200); // wait 200 milliseconds
}
}
100 %
Output
Show output from: Build
Target "Build" in file "C:\Program Files\Atmel\AVR Studio 5.0\Vs\Avr.common.t
Done building target "Build" in project "firstprogram.avrgccproj".
Done building project "firstprogram.avrgccproj".
Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========


firstprogram - AVR Studio
File Edit View VAssistX Project Build Debug Tools Window Help
AVR Tools Firmware Upgrade
AVR Programming
Code Snippets Manager... Ctrl+K, Ctrl+B
Add-in Manager...
Extension Manager...
External Tools...
Import and Export Settings...
Customize...
Options...
firstprogram.c
#include <avr/io.h>
#include <avr/delay.h> // has some de
int main(void)
{
DDRC=0xff; // all C ports as outp
while(1)
{
//TODO:: Please write your application code
PORTC=0xff; // turn on all C porst
_delay_ms(200); // wait 200 milliseconds
PORTC=0x00; // turn off all c ports
_delay_ms(200); // wait 200 milliseconds


AVR Programming
Tool
Device
Interface
Device ID
Target Voltage
AVRISP mkII
ATmega328P
ISP
Apply
not read
Read

Select tool, device and interface.
Close





hex file checkbox, make sure it's checked.

Now to actually generate the hex file click Build->Build Solution avr studio build menuthis will compile the program and generate the hex file we need. You should not get any errors, as shown below.

Program compilation errors no compilation errors Upload The Code to The Microcontroller

At last, we are done with all the software developing stuff, uploading is just as easy. Click on Tools->AVR Programming avr studio tools menu

You'll get the programming window. Make sure that your device is selected.

avr programming window the programming window Now in the following order, click on the Apply button, then the voltage button (recycling/refresh looking icon) will become active now click it, you should get some value close to 5 volts. Then click on the Read button. If you didn't get any error window you are good to go.

Clicking the Read and Voltage buttons is not necessary actually, but the Apply button is.avr studio programming window: 5 volts and device id programming window after clicking buttonslastly click on Memories and select your hex file then click Program, the programmer will start blinking and stop when your program is done uploading.avr studio memories window, hex file ready to upload hex file ready for upload

Coding:

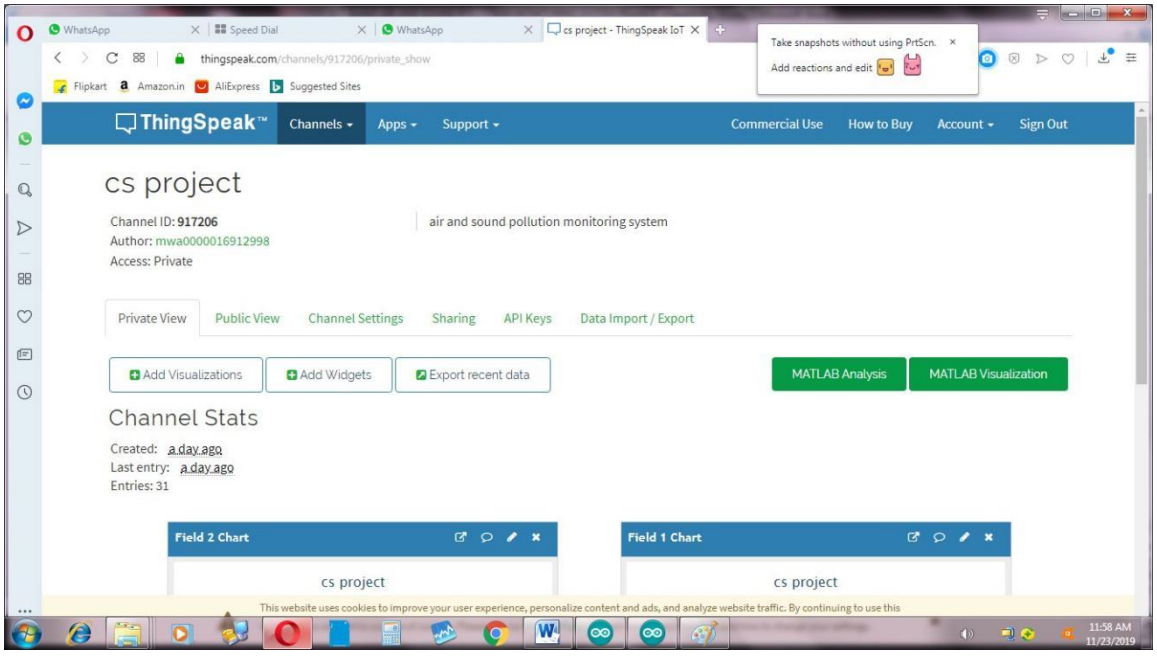
```
#include "config.h"
#include <Wire.h>

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

/******** Example Starts Here **********/

// analog pin 0
#define PHOTOCELL_PIN A0
const int soundsensor=D5;
int sensorstate=1;
// photocell state
```

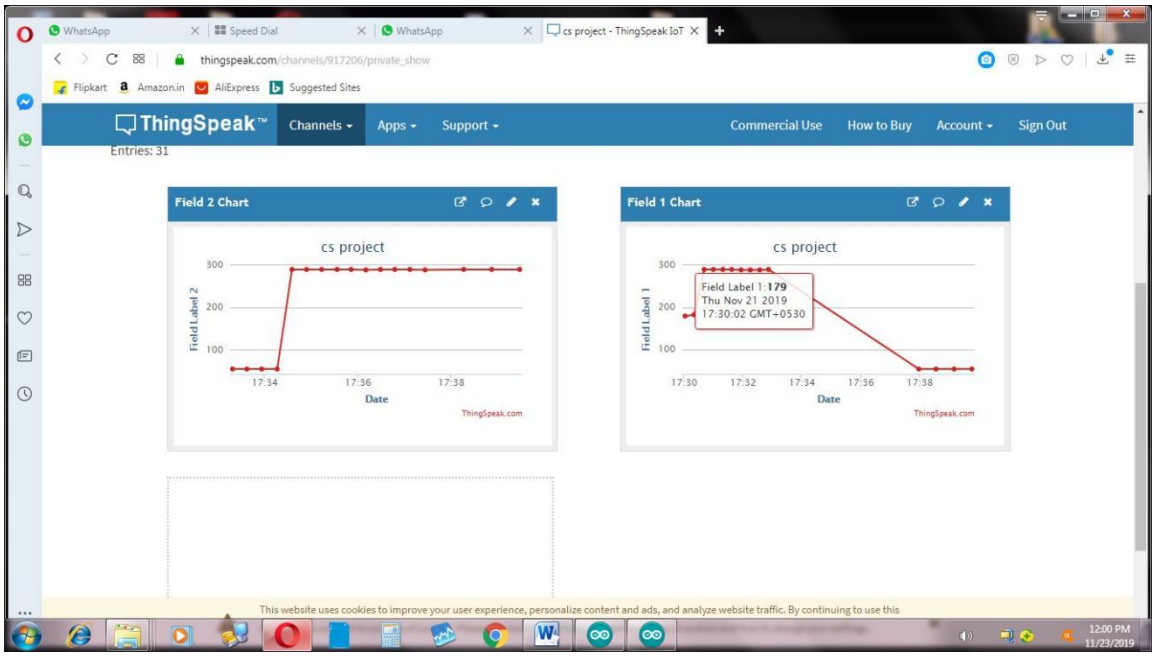
```
int current = 0;
int last = -1;
// set up the 'analog' feed
AdafruitIO_Feed *analog = io.feed("air");
AdafruitIO_Feed *digital = io.feed("sound");
AdafruitIO_Feed *digital1 = io.feed("display");
void setup() {
  // start the serial connection
    Serial.begin(115200);
       lcd.init();
       lcd.backlight();
       lcd.clear();
       lcd.setCursor(0,0);
  lcd.print("AIR AND SOUND");
lcd.setCursor(0,1);
lcd.print("POLLUTION..");
 delay(1500);
  // wait for serial monitor to open
  while(! Serial);

  // connect to io.adafruit.com
  Serial.print("Connecting to Adafruit IO");
  io.connect();

  // wait for a connection
  while(io.status() < AIO_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
```

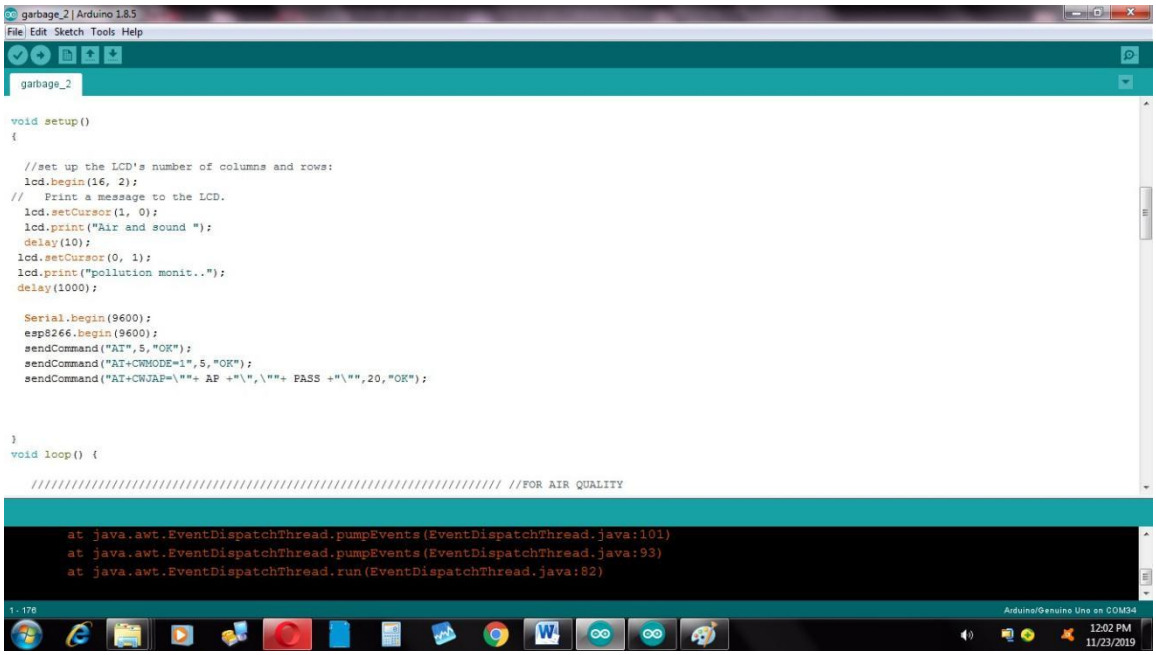
```
  // we are connected
  Serial.println();
  Serial.println(io.statusText());

}

void loop() {

  // io.run(); is required for all sketches.
  // it should always be present at the top of your loop
  // function. it keeps the client connected to
  // io.adafruit.com, and processes any incoming data.
  io.run();

  // grab the current state of the photocell
  current = analogRead(PHOTOCELL_PIN);
sensorstate=digitalRead(soundsensor);
  // return if the value hasn't changed
  if(current == last)
    return;

  // save the current state to the analog feed
  Serial.print("sending -> ");
  Serial.println(current);
  analog->save(current);

  // store last photocell state
  last = current;

  // wait three seconds (1000 milliseconds == 1 second)
  //
```

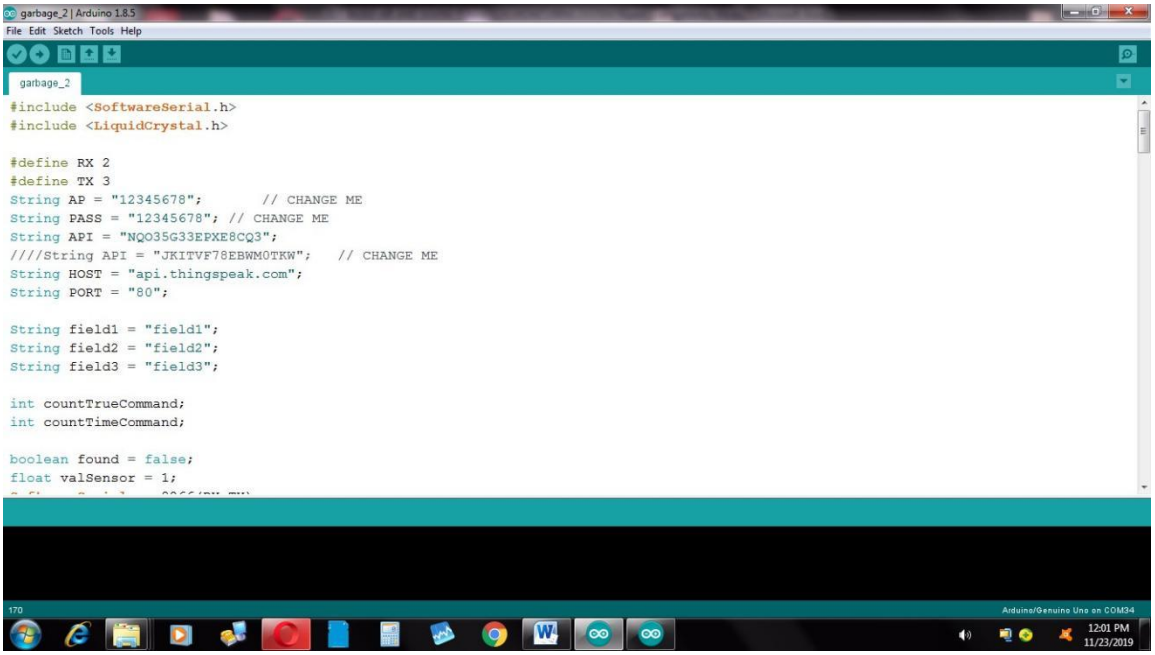
```
  // because there are no active subscriptions, we can use delay()
  // instead of tracking millis()
  lcd.clear();
      lcd.setCursor(0,0);
  lcd.print("AIR");
lcd.setCursor(0,1);
lcd.print(current);
  lcd.setCursor(7,0);
  lcd.print("Sound");
lcd.setCursor(7,1);
lcd.print("No");

  delay(2000);

if(current>280){

  lcd.clear();
      lcd.setCursor(0,0);
  lcd.print("Air pollution");
lcd.setCursor(0,1);
lcd.print("detected");

  digital1->save("Air Pollution detected");
  delay(3000);
  }
  if(sensorstate==HIGH){

     lcd.clear();
      lcd.setCursor(0,0);
  lcd.print("Sound pollution");
lcd.setCursor(0,1);
```

```
lcd.print("detected");

  digital1->save("Sound POLLUTION DETECTED");
  delay(3000);
    }
    if(sensorstate==LOW){

       digital1->save("MONITORING AIR AND SOUND POLLUTION");
  delay(3000);
    }
}
```




ThingSpeak
Channels
Apps
Support
Commercial Use
How to Buy
Account
Sign Out
Entries: 31
Field 2 Chart
cs project
Field Label 2
Date
Field 1 Chart
cs project
Field Label 1
Field Label 1: 179
Thu Nov 21 2019
17:30:02 GMT+0530
Date
This website uses cookies to improve your user experience, personalize content and ads, and analyze website traffic. By continuing to use this


garbage_2 | Arduino 1.8.5
File Edit Sketch Tools Help
garbage_2
void setup()
{
//set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.setCursor(1, 0);
lcd.print("Air and sound ");
delay(10);
lcd.setCursor(0, 1);
lcd.print("pollution monit..");
delay(1000);
Serial.begin(9600);
esp8266.begin(9600);
sendCommand("AT",5,"OK");
sendCommand("AT+CWMODE=1",5,"OK");
sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK");
}
void loop() {
// //FOR AIR QUALITY
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)
1 - 176
Arduino/Genuino Uno on COM34


garbage_2 | Arduino 1.8.5
File Edit Sketch Tools Help
garbage_2
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#define RX 2
#define TX 3
String AP = "12345678";   // CHANGE ME
String PASS = "12345678"; // CHANGE ME
String API = "NQO35G33EPXE8CQ3";
////String API = "JKITVF78EBWM0TKW";   // CHANGE ME
String HOST = "api.thingspeak.com";
String PORT = "80";
String field1 = "field1";
String field2 = "field2";
String field3 = "field3";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
float valSensor = 1;
170
Arduino/Genuino Uno on COM34
12:01 PM
11/23/2019

Chapter 7

RESUIT DISCUSSION

In this chapter we have to discuss about the result of model and here the some result screenshot is mentioned below

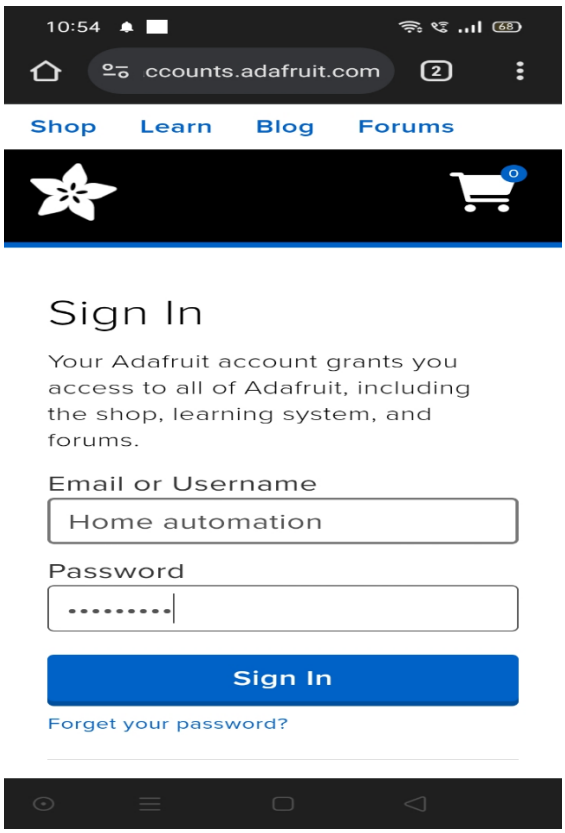


Fig 7.1 Screenshot of hotspot connection to device

Fig 7.2 dashboard of air and noise pollution monitoring by iot

Fig 7.3 after Hotspot connection it initially display air and sound pollution

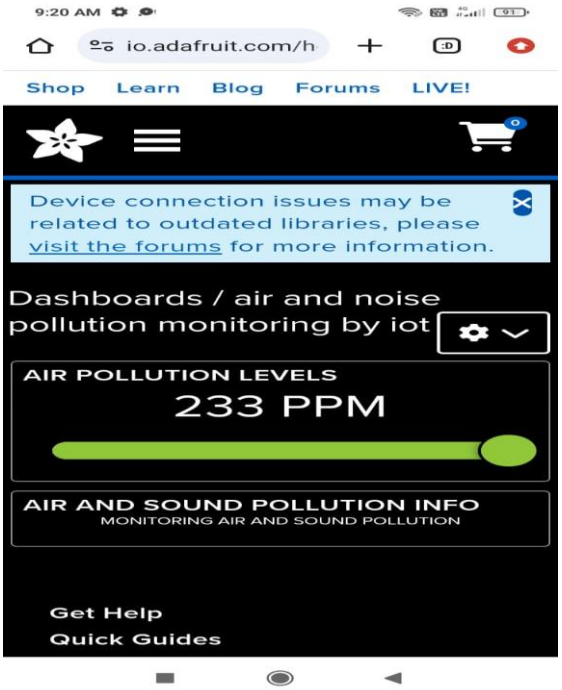


Figure 7.5 above the set point value of sound is received by microphone then it show

Fig 7.4 after the threshold value of air, it shows the air pollution detected

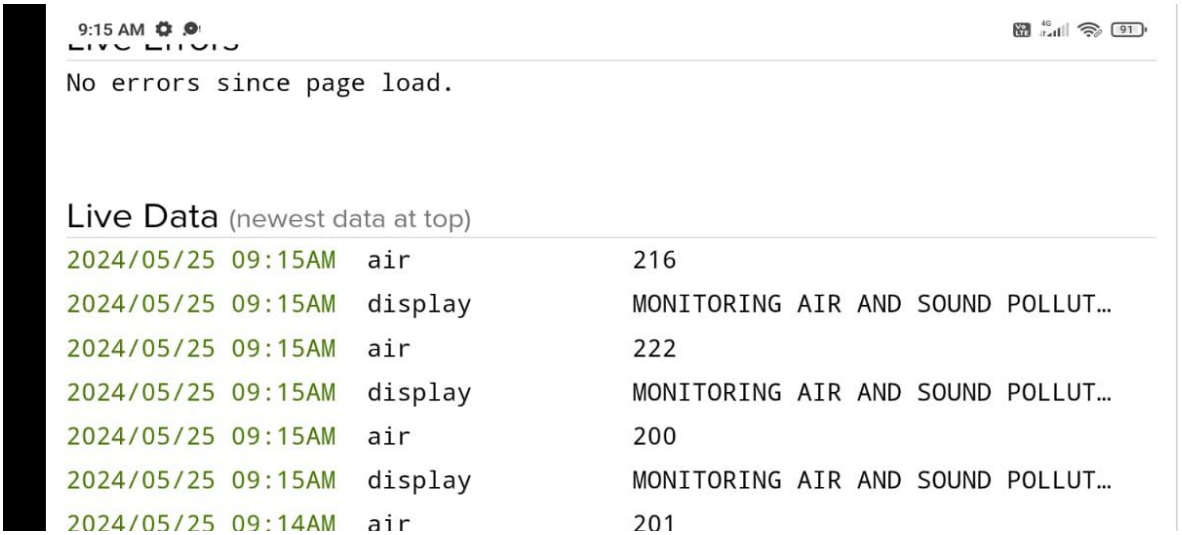


Fig 7.6 screen shot for live data verification on the dashboard

Chapter 8

ADVANTAGES AND APPLICATIONS

Advantages:

- ✓ The system can be used to monitor air quality and sound.
- ✓ Data can used in researches and data mining.
- ✓ It is easy to use safe and reliable.
- ✓ Sensors are easily available.
- ✓ Sensors are effortlessly accessible.
- ✓ Detecting of wide range of gases.
- ✓ Simple, compact and easy to handle.
- ✓ Sensors have long life time.
- ✓ Low cost
- ✓ Data can be used to control pollution

Application:

- ✓ To estimate the pollution.
- ✓ Indoor Air Quality Monitoring.
- ✓ To design server and upload data on that server with date and time.
- ✓ We can use it at industrial area as there is lot of noise pollution
- ✓ In city roads traffic noise.
- ✓ Activities like shooting, open air events, football and cricket matches.
- ✓ At small level, in schools and colleges we can use this device.

Chapter 9

CONCLUSION AND FUTURE SCOPE

By using this project each and every variation we can analyze and inform nearby people in time. We can also analyze data form home using thing speak. The most important factor of this system is that it is small, cost efficient and portable. Sensors are available easily anywhere. This system fully helpful to save the lives and overcome all the problem related to environment.

Future scope:

- ✓ In future we will add more accurate sensors for accurate analysis of data.
- ✓ We will work on the power supply and replace existing system with solar based power supply.
- ✓ We can monitor air and sound pollution level at any place of the world.

REFERENCES

1] Augarten, Stan (1983). The Most Widely Used Computer on a Chip: The TMS 1000. State of the Art: A Photographic History of the Integrated Circuit (New Haven and New York: Ticknor & Fields). ISBN 0-89919-195-9.Retrieved 2009- 12-23.

2] "Oral History Panel on the Development and Promotion of the Intel 8048 Microcontroller" (PDF). *Computer History Museum Oral History, 2008*.p. 4.Retrieved 2011-06-28.

3] "Atmel's Self-Programming Flash Microcontrollers" (PDF). 2012-01-24. Retrieved 2008-10-25.by Odd Jostein Svendsli 2003

4] Jim Turley. "The Two Percent Solution" 2002.

5] Tom Cantrell "Microchip on the March". Circuit Cellar. 1998.http://www.semico.com

6] Momentum Carries MCUs Into 2011 http://semico.com/content/momentum- carries-mcus-2011

7] "MCU Market on Migration Path to 32-bit and ARM-based Devices". April 25, 2013. It typically takes a global economic recession to upset the diverse

8] IOT based Air and Sound Pollution Monitoring System International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Website: www.ijareeie.com Vol. 6, Issue 3, March 2017 Copyright to IJAREEIE

9] IoT Based Air And Sound Pollution Monitoring System International Journal Of Innovative Research Explorer Volume 5, Issue 4, April/2018 Issn No: 2347-6060

10] Design And Analysis Of IoT Based Air Quality Monitoring System 2020 International Conference On Power Electronics &IoT Applications In Renewable Energy And Its Control (PARC) GLA University, Mathura, UP, India. Feb 28-29, 2020

11] S Dhingra, RbMadda, Ah Gandomi - Ieee Internet Of Things (IoT) Is A Worldwide System Of "Smart Devices" That Can Sense And Connect With Their Surroundings And Interact With Users And Other Systems. Global Air Pollution Is One Of The Major Concerns Of Our EraHttps://Ieeexplore.Ieee.Org/Abstract/Document/86633 67

12] KB Shaban A Kadri, E Rezk - IEEE Sensors Journal, A System For Monitoring And Forecasting Urban Air Pollution Is Presented In This Paper. The System Uses Low-Cost Air-Quality Monitoring Motes That Are Equipped With An Array Of Gaseous And

Meteorological Sensors.

13] Air and Sound Pollution Monitoring System Using IOT International Research Journal of Engineering and Technology (IRJET e-ISSN: 2395-0056

14] P.Sai Chandana , K.Sreelekha , A.Muni Likith Reddy , M.Anil KumarReddy ,R.Senthamilselvan "IOT Air And Sound Pollution Monitoring System," International Journal on Applications in Engineering and Technology, Volume 3: Issue 1: March 2017.

15] P.Vijnatha Raju, R.V.R.S.Aravind, B Sangeeth Kumar, "Pollution Monitoring System using Wireless Sensor Network" IJETT,Volume4,Issue4-April 2013.

16] Godbless Swararya, Shubi Kaijage, Ramasdhani S.Sinde- Electronics and Telecommunication Department, " Air Pollution Monitoring System Based On Wireless Networks – Simulation" ISSN , Vol 5 , No 8 ,2014.

17] Jadhav Aditya .S ,Pawar Vishwajeet. P, Jorwekar Sagar .R ,Jadhav Vidya .P, Computer Engineering ,India , " Industrial Air Pollution Monitoring And Analysis System." ISSN ,Vol 6 ,Issue 3, March 2016.