**Noise pollution Monitoring System**

**Using ESP8266**

**CODE:**

#define BYYNK\_PRINT Serial

#include <ESP8277Wifi.h>

#include <BlynkSimpleEsp8266.h>

#include<LiquidCrystal\_I2C.h>

#define SENSOR\_PIN A0

LiquidCrustal\_I2C Icd(0x3F,2,1,0,4,5,6,7,3, POSITIVE);

Const int sampleWindow = 50;

Unsigned int sample;

Int db;

Char auth[]=“IEuIxT825VDt6hNfrcFgdj6inj1QUfsA”;

Char ssid{}=“real 6”;

Char pass[]=“evil@zeb”;

BLYNK\_READ(V0)

{

Blynk.virtualWrite(V0,db);

}

Void setuo() {

pinMode(SENSOR\_PIN, INPUT);

Icd.begin916,2);

*Icd..backlight();*

*Icd.clear();*

*Blynk.begin(auth, ssid,pass);*

*}*

*Void loop() {*

*Blynk.run();*

*unsigned long startMillis = millis(); // Start of sample window*

*float peakToPeak = 0; // peak-to-peak level*

*unsigned int signalMax = 0; //minimum value*

unsigned int signalMin = 1024; //maximum value

// collect data for 5- mS

While (millis() – startMillis < sampleWindow)

{

sample = analogRead()SENSOR\_PIN); //get reading from microphone

if (sample < 1024) // toss out spurious readings

{  
 sample = analogRead(SENSOR\_PIN); //get reading from microphone  
 if (sample<1024) // toss out spurious readings  
 {  
 if (sample > signalMax)  
 {  
 signalMax = sample; // save just the max levels  
 }  
 else if (sample < signalMin)  
 {   
 signalMin = sample; // save just the min levels  
 }   
 }

peadToPeak = signalMax – signalMin; // max-min = pead-peak amplitude  
 Serial.printIn(peakToPeak);  
 db = map(peadToPeak, 20, 900.49.5,90); //calibrate for deciBels   
 Icd.print(“Loudness;’);  
 Icd.print(db);  
 Icd.print(‘dB”);  
 if(db <=50)  
 {  
 Icd.setCursor(0,1);  
 Icd.print(“Level: Quite”)l  
  
 }  
 else if (db>50 && db<75)  
 {  
 Icd.setCursor(0,1);  
 Icd.print(“Level: Moderate”);  
 }  
 else if(db>=75)  
 {  
 Icd.setCursor(0,1);  
 Icd.print(“Level: high”);  
 }  
 delay(600);  
 Icd.clear();  
 }