**Arduino Codes for KIONA: Smart Society Automation**

1. **Code for TFT Screen, Turbidity sensor and TDS sensors using Arduino Mega Board:**

#include<Adafruit\_GFX.h>

#include<**MCUFRIEND\_kbv**.h>

#define BLACK   0x0000

#define BLUE    0x001F

#define LIGHTGREY   0xC618

#define RED     0xF800

#define GREEN   0x07E0

#define CYAN    0x07FF

#define MAGENTA 0xF81F

#define YELLOW  0xFFE0

#define WHITE   0xFFFF

#define TFT\_WIDTH  480

#define TFT\_HEIGHT 320

#define COLUMN\_WIDTH (TFT\_WIDTH / 2)

#define ROW\_HEIGHT (TFT\_HEIGHT / 7)

//(int CS=A3, int Rs=A2, int WR=A1, int RD=A0, int RST=A4)

**MCUFRIEND\_kbv** tft(A3, A2, A1, A0, A4);

int sensorValue=0;

float turbidity = 0.00;

float Vclear=2.85; // output voltage to calibrate (with clear water)

//TDS1

int tds1\_sensor=A14;

float aref1 = 4.3;

float ecCalibration1 = 1;

float ec1 = 0;

unsigned int tds1 = 0;

int raw\_temp1;

float waterTemp1 = 0;

//TDS2

int tds2\_sensor=A15;

float aref2 = 4.3;

float ecCalibration2 = 1;

float ec2 = 0;

unsigned int tds2 = 0;

int raw\_temp2;

float waterTemp2 = 0;

void setup()

{

**Serial**.begin(9600);

 uint16\_t ID= tft.readID();

 tft.begin(ID);

 tft.invertDisplay(true);

 tft.setRotation(1);

 //TDS1

 pinMode(tds1\_sensor, INPUT);

**Serial**.begin(9600);

 //TDS2

 pinMode(tds2\_sensor, INPUT);

**Serial**.begin(9600);

}

void loop(void)

{

 //Code for Turbidity Sensor

 //Read the analog value from sensor

 sensorValue= analogRead(A13);

 //convert the analog value to a voltage.

 float voltage = sensorValue\*(5.0/1024.0);

 //calculate the turbidity

 turbidity = 100.00 -(voltage/Vclear)\*100.00;

//TDS1

float rawEc1 = analogRead(tds1\_sensor) \* aref1 / 1024.0;

 float temperatureCoefficient1 = 1.0 + 0.02 \* (waterTemp1 - 25.0);

 ec1 = (rawEc1 / temperatureCoefficient1) \* ecCalibration1;

 tds1 = (133.42 \* pow(ec1, 3) - 255.86 \* ec1 \* ec1 + 857.39 \* ec1) \* 0.5;

  //TDS2

float rawEc2 = analogRead(tds2\_sensor) \* aref2 / 1024.0;

 float temperatureCoefficient2 = 1.0 + 0.02 \* (waterTemp2 - 25.0);

 ec2 = (rawEc2 / temperatureCoefficient2) \* ecCalibration2;

 tds2 = (133.42 \* pow(ec2, 3) - 255.86 \* ec2 \* ec2 + 857.39 \* ec2) \* 0.5;

// side 1: colls

 tft.fillRect(0,0,480,64,BLUE);

  tft.fillRect(0,60,480,64,BLACK);

 tft.fillRect(0,63,240,64,BLACK);

 tft.fillRect(0,65,240,64,WHITE);

  tft.fillRect(0,113,240,64,BLACK);

   tft.fillRect(0,116,240,64,WHITE);

 tft.fillRect(0,164,240,64,BLACK);

 tft.fillRect(0,167,240,64,WHITE);

 tft.fillRect(0,215,240,64,BLACK);

  tft.fillRect(0,218,240,64,WHITE);

 tft.fillRect(0,266,240,64,BLACK);

  tft.fillRect(0,269,240,64,CYAN);

//2nd Side

  //tft.fillRect(241,0,240,64, GREEN);

 tft.fillRect(241,63,240,64, BLACK);

  tft.fillRect(241,65,240,64, WHITE);

 tft.fillRect(241,113,240,64, BLACK);

 tft.fillRect(241,116,240,64, WHITE);

 tft.fillRect(241,164,240,64, BLACK);

  tft.fillRect(241,167,240,64, WHITE);

 tft.fillRect(241,215,240,64, BLACK);

  tft.fillRect(241,218,240,64, WHITE);

 tft.fillRect(241,266,240,64,BLACK);

  tft.fillRect(241,269,240,64,CYAN);

// Heading:

tft.setTextColor(WHITE);

 tft.setTextSize(2.9);

tft.setCursor(50, 0+20);

tft.print("KIONA:WATER PURIFICATION SYSTEM");

 //TDS1\_PPM

 tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

 tft.setCursor(20, 65+20);//50, 0+20

 tft.print("TDS\_PRE\_PPM: ");

//delay(5000);

 //TDS1\_EC

 tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

 tft.setCursor(20, 116+20);

 tft.print("TDS\_PRE\_EC: ");

 //delay(5000);

 //TDS2\_PPM

 tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

 tft.setCursor(20, 167+20);

 tft.print("TDS\_POST\_PPM: ");

 //delay(5000);

 //TDS2\_EC

 tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

 tft.setCursor(20, 218+20);

 tft.print("TDS\_POST\_EC: ");

 //

 //Turbidity

 tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

 tft.setCursor(20, 269+20);

 tft.print("TURBIDITY: ");

 //on the screen: print values

 //TDS1\_PPM

 tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

 tft.setCursor(260, 65+20);

 tft.print(tds1);

 //tft.print((char)247);

 tft.println(" PPM");

//TDS1\_EC

tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

tft.setCursor(260, 116+20);

tft.print(ec1, 2);

tft.println(" mS/cm");

// TDS2\_PPM

tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

tft.setCursor(260, 167+20);

tft.print(tds2);

tft.println(" PPM");

//TDS2\_EC

tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

tft.setCursor(260, 218+20);

tft.print(ec2,2);

tft.println(" mS/cm");      // unit

// Turbidity

tft.setTextColor(BLACK);

 tft.setTextSize(2.5);

tft.setCursor(260, 269+20);

tft.print(turbidity);

tft.println(" NTU");

  delay (20000);

 // on serial monitor: Turbidity

**Serial**.print("Turbidity= ");

**Serial**.print(turbidity);

 //Serial.print(" NTU");

**Serial**.println();

// on Serial Monitor : TDS1\_PPM

**Serial**.print("TDS1\_PPM =");

 //Serial.print("TDS1\_ppm");

 //Serial.print("ppm");

**Serial**.println();

 // on Serial Monitor : TDS1\_EC

**Serial**.print("TDS1\_EC =");

 //Serial.print("TDS1\_EC");

 //Serial.print("unit");

**Serial**.println();

 // on Serial Monitor : TDS2\_PPM

**Serial**.print("TDS2\_PPM =");

 //Serial.print("TDS2\_ppm");

 //Serial.print("ppm");

**Serial**.println();

// on Serial Monitor : TDS1\_EC

**Serial**.print("TDS2\_EC =");

 //Serial.print("TDS2\_EC");

 //Serial.print("unit");

**Serial**.println();

 delay(500);

}

1. **Code for 16 x 2 I2C LCD Display and Gas Sensor using Arduino UNO Board:**

#include <**LiquidCrystal\_I2C**.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display

**LiquidCrystal\_I2C** lcd(0x27,16,2);

#define **Buzzer** 2

#define RED 3

#define GREEN 5

void setup() {

  lcd.init();

 lcd.clear();

 lcd.backlight(); // Turns on backlight

 pinMode(A1, INPUT);

 pinMode(RED, OUTPUT);

 pinMode(GREEN, OUTPUT);

 pinMode(**Buzzer**, OUTPUT);

**Serial**.begin(9600);

 // initialize the LCD

 lcd.begin(16,2);

}

void loop() {

int gas= analogRead(A1);

**Serial**.println(gas);

//Serial.print("GAS");

lcd.setCursor(0,0);

//lcd.print("Hazards System");

//delay(400);

if(gas > 500)

{

 digitalWrite(RED,HIGH);

 digitalWrite(GREEN, LOW);

 digitalWrite(**Buzzer**, HIGH);

 lcd.clear();

 lcd.setCursor(1,0);

 lcd.print("CALL TO RESCUE");

 lcd.setCursor(4,1);

 lcd.print("SERVICES");

}

else

{

 digitalWrite(GREEN,HIGH);

 digitalWrite(RED, LOW);

 digitalWrite(**Buzzer**, LOW);

 lcd.clear();

 lcd.setCursor(2,0);

 lcd.print("KIONA : HMS");

}

delay(200);

}

**I2C Board of LCD Arduino**

|  |  |
| --- | --- |
| **I2C LCD Display** | **Arduino UNO Board** |
| **GND** | **GND** |
| **VCC** | **5V/ Vin** |
| **SDA** | **A4** |
| **SCL** | **A5** |