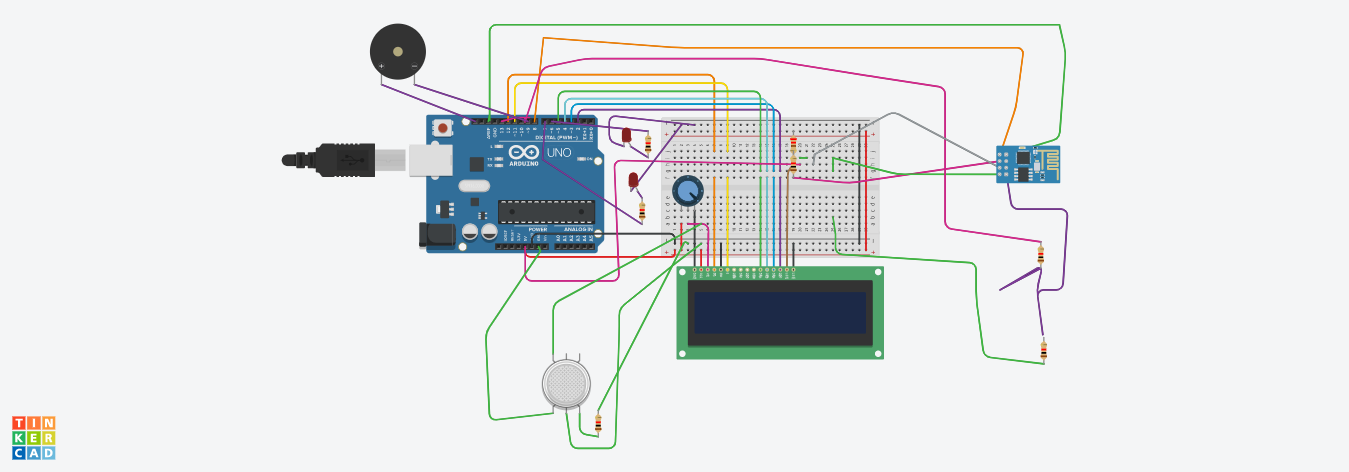
***SMOKE ALARAM SYSTEM***

C

C CODE FOR SMART ALARAM SYSTEM

|  |
| --- |
| #include <SoftwareSerial.h> |
|  | #include <LiquidCrystal.h> |
|  |  |
|  | // initialize the library with the numbers of the interface pins |
|  | LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // LCD Connections |
|  | SoftwareSerial SerCommESP8266(8,9); // RX, TX connect 8 to TX of ESP, connect 9 to RX of ESP |
|  |  |
|  | int smokeVal=0; |
|  | int smoke\_sensor\_pin=A0; // MQ2 Gas Sensor |
|  | int red\_led\_pin=7; // Smoke indication |
|  | int green\_led\_pin=6; // No Smoke indication |
|  | int buzzer\_pin = 10; // Buzzer |
|  |  |
|  | String apiKey = "2TDYYE99BAABM1P8"; // Write API key |
|  |  |
|  | void setup() |
|  | { |
|  | pinMode(red\_led\_pin, OUTPUT); |
|  | pinMode(green\_led\_pin, OUTPUT); |
|  | pinMode(buzzer\_pin, OUTPUT); |
|  | pinMode(smoke\_sensor\_pin, INPUT); |
|  | Serial.begin(9600); // serial data transmission at Baudrate of 9600 |
|  | SerCommESP8266.begin(9600); // Initialize the serial communication baud rate |
|  |  |
|  | lcd.begin(16, 2); // to intialize LCD |
|  | lcd.setCursor(0,0); |
|  | lcd.print(" Welcome"); |
|  | lcd.setCursor(0,1); |
|  | lcd.print(" To "); |
|  | delay(1000); |
|  | lcd.clear(); |
|  | lcd.setCursor(0,0); |
|  | lcd.print(" Technical"); |
|  | lcd.setCursor(0,1); |
|  | lcd.print(" Update"); |
|  | delay(1000); |
|  | SerCommESP8266.println("AT"); // Start ESP8266 Module |
|  | delay(1000); |
|  | SerCommESP8266.println("AT+GMR"); // To view version info for ESP-01 output: 00160901 and ESP-12 output: 0018000902-AI03 |
|  | delay(1000); |
|  | SerCommESP8266.println("AT+CWMODE=3"); // To determine WiFi mode |
|  | delay(1000); |
|  | SerCommESP8266.println("AT+RST"); // To restart the module |
|  | delay(1000); |
|  | SerCommESP8266.println("AT+CIPMUX=1"); // Enable multiple connections 0: Single connection 1: Multiple connections (MAX 4) |
|  |  |
|  | delay(1000); |
|  | String cmd="AT+CWJAP=\"SSID NAME\",\"SSID PASSWORD\""; // connect to Wi-Fi |
|  | SerCommESP8266.println(cmd); |
|  | delay(1000); |
|  | SerCommESP8266.println("AT+CIFSR"); // Return or get the local IP address |
|  | delay(1000); |
|  | lcd.clear(); |
|  | lcd.setCursor(0,0); |
|  | lcd.print(" WIFI"); |
|  | lcd.setCursor(0,1); |
|  | lcd.print(" CONNECTED"); |
|  | } |
|  |  |
|  |  |
|  | void loop() |
|  | { |
|  | delay(1000); |
|  | smokeVal = map(analogRead(A0),10,350,0,100); |
|  | Serial.println(); |
|  | lcd.clear(); |
|  | lcd.setCursor (0, 0); |
|  | lcd.print (smokeVal); |
|  | lcd.print (" In Room"); |
|  | lcd.setCursor (0,1); |
|  | if (smokeVal>30) |
|  | { |
|  | lcd.print("Smoke Detected"); |
|  | Serial.print("Smoke Detected"); |
|  | digitalWrite(red\_led\_pin, HIGH); |
|  | digitalWrite(green\_led\_pin, LOW); |
|  | tone(buzzer\_pin, 1000, 200); |
|  | } |
|  | else |
|  | { |
|  | lcd.print("Safe"); |
|  | Serial.print("Safe"); |
|  | digitalWrite(red\_led\_pin, LOW); |
|  | digitalWrite(green\_led\_pin, HIGH); |
|  | noTone(buzzer\_pin); |
|  | } |
|  | delay(1000); |
|  | lcd.clear(); |
|  | lcd.setCursor(0,0); |
|  | lcd.print(" SENDING DATA"); |
|  | lcd.setCursor(0,1); |
|  | lcd.print(" TO CLOUD"); |
|  | SetupESP8266\_HA(); // For ThingSpeak Data Transfer |
|  | delay(1000); |
|  | } |
|  |  |
|  | void SetupESP8266\_HA() |
|  | { |
|  | // TCP connection AT+CIPSTART=4,"TCP","184.106.153.149",80 |
|  | String cmd = "\nAT+CIPSTART=4,\"TCP\",\""; // Establish TCP connection |
|  | cmd += "184.106.153.149"; // api.thingspeak.com |
|  | cmd += "\",80"; // Port Number |
|  | SerCommESP8266.println(cmd); |
|  | Serial.println(cmd); |
|  | if(SerCommESP8266.find("Error")) |
|  | { |
|  | Serial.println("AT+CIPSTART error"); |
|  | return; |
|  | } |
|  | String getStr = "GET /update?api\_key="; // API key |
|  | getStr += apiKey; |
|  | getStr +="&field1="; // Field variable as Smoke |
|  | getStr +=String(smokeVal); |
|  | getStr += "\r\n\r\n"; |
|  | // send data length |
|  | cmd = "AT+CIPSEND="; // Send data AT+CIPSEND=id,length |
|  | cmd += String(getStr.length()); |
|  | SerCommESP8266.println(cmd); |
|  | Serial.println(cmd); |
|  | delay(1000); |
|  | SerCommESP8266.print(getStr); |
|  | Serial.println(getStr); |
|  | // thingspeak needs max 16 sec delay between updates |
|  | delay(10000); |
|  | }  LINK:  https://www.tinkercad.com/things/8q7fV1pCVyG-copy-of-wifi-module-esp8266/editel?sharecode=c0p\_VyoaCKUg\_E1xXxFNcg-SRRtSS-NS73zsVCUbYGY |

**Footer**

© 2022 GitHub, Inc.

**Footer navigation**

* [Terms](https://docs.github.com/en/github/site-policy/github-terms-of-service)
* [Privacy](https://docs.github.com/en/github/site-policy/github-privacy-statement)
* [Security](https://github.com/security)
* [Status](https://www.githubstatus.com/)
* [Docs](https://docs.github.com/)
* [Contact GitHub](https://support.github.com/?tags=dotcom-footer)
* [Pricing](https://github.com/pricing)
* [API](https://docs.github.com/)
* [Training](https://services.github.com/)
* [Blog](https://github.blog/)
* [About](https://github.com/about)