|  |  |
| --- | --- |
| **Smart Mining Helmet Using 8051 Microcontroller** | **Abstract**  **This project aims to develop a Smart Mining Helmet equipped with advanced sensors and a microcontroller to enhance safety measures by monitoring environmental conditions such as gas levels, temperature, humidity, and proximity.**  .**VIVEK NAIK**  **EMBEDDED SYTEM** |

**Smart Mining Helmet Using 8051 Microcontroller**

**AIM:**

**The aim of the Smart Mining Helmet project utilizing the 8051 Microcontroller is to enhance mining safety and efficiency by integrating advanced technology into traditional mining practices. This innovative helmet aims to provide real-time monitoring of environmental parameters such as gas levels, temperature, and humidity, ensuring the well-being of miners. Additionally, the project focuses on incorporating features like location tracking and communication capabilities to enable quick response in case of emergencies. By seamlessly integrating smart functionalities into the mining helmet, the project to reduce risks, enhance operational effectiveness, and ultimately contribute to a safer and more productive mining environment.**

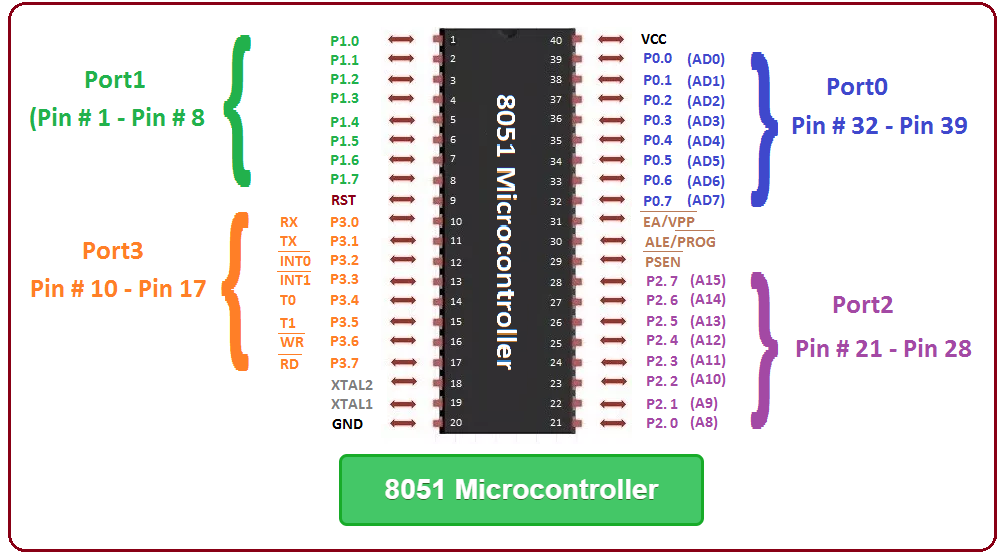
**DESIGN:**

**The Smart Mining Helmet designed with the 8051 Microcontroller integrates various sensors to enhance safety and monitoring capabilities in mining environments. The Ultrasonic Sensor is employed for proximity detection, ensuring miners are alerted to potential obstacles or hazards in their vicinity. The DHT11 Sensor measures temperature and humidity, providing crucial environmental data for miners' well-being. An LDR (Light Dependent Resistor) sensor is incorporated to monitor ambient light conditions, ensuring optimal visibility. The MQ-5 Gas Sensor enhances safety by detecting and monitoring gas levels in real-time, mitigating the risks associated with gas exposure.**

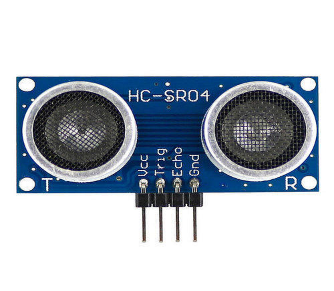
**These sensors are seamlessly interfaced with the 8051 Microcontroller, which acts as the central processing unit for data acquisition and control. The microcontroller processes information from the sensors and initiates appropriate responses, such as sounding alarms or sending alerts, in the event of abnormal conditions. This comprehensive sensor integration ensures a holistic approach to mining safety, combining environmental monitoring with immediate actionable insights for miners, thereby fostering a safer and more secure mining environment.**

**REQUIREMENT:**

**8051 Microcontroller:**

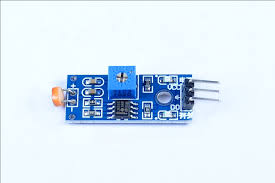
****

**The central processing unit responsible for data processing, decision-making, and controlling various components of the smart mining helmet.**

**Ultrasonic Sensor:**

* **Monitors and records ambient temperature and humidity levels within the mining environment.**
* **Provides crucial data for assessing comfort conditions and ensuring the well-being of miners.**

**LDR Sensor:**

****

* + **Measures ambient light levels in the mining area.**
  + **Enables automatic adjustment of helmet-mounted lighting, enhancing visibility and safety in low-light conditions.**

** MQ5 Gas Senso**

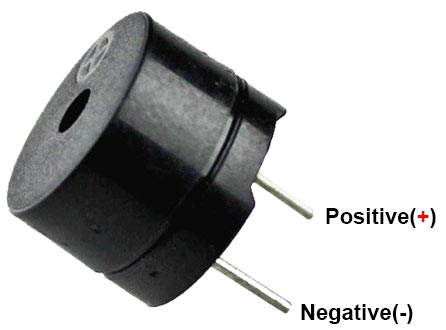
* **Detects the presence of harmful gases such as methane and carbon monoxide.**
* **Ensures real-time monitoring of air quality, preventing potential health hazards for miners.**

**User Interface (LED/LCD Display):**

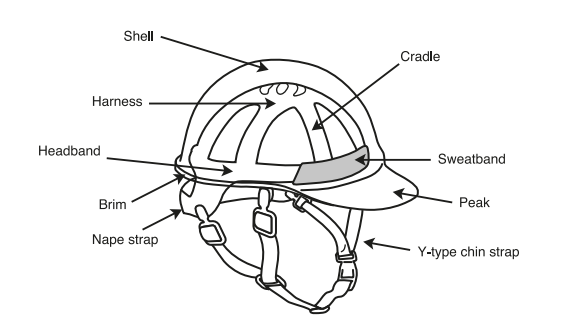
****

* **Interfaces with the microcontroller to display real-time data, warnings, and alerts to the miner.**
* **Enhances user interaction and provides immediate feedback on environmental conditions.**

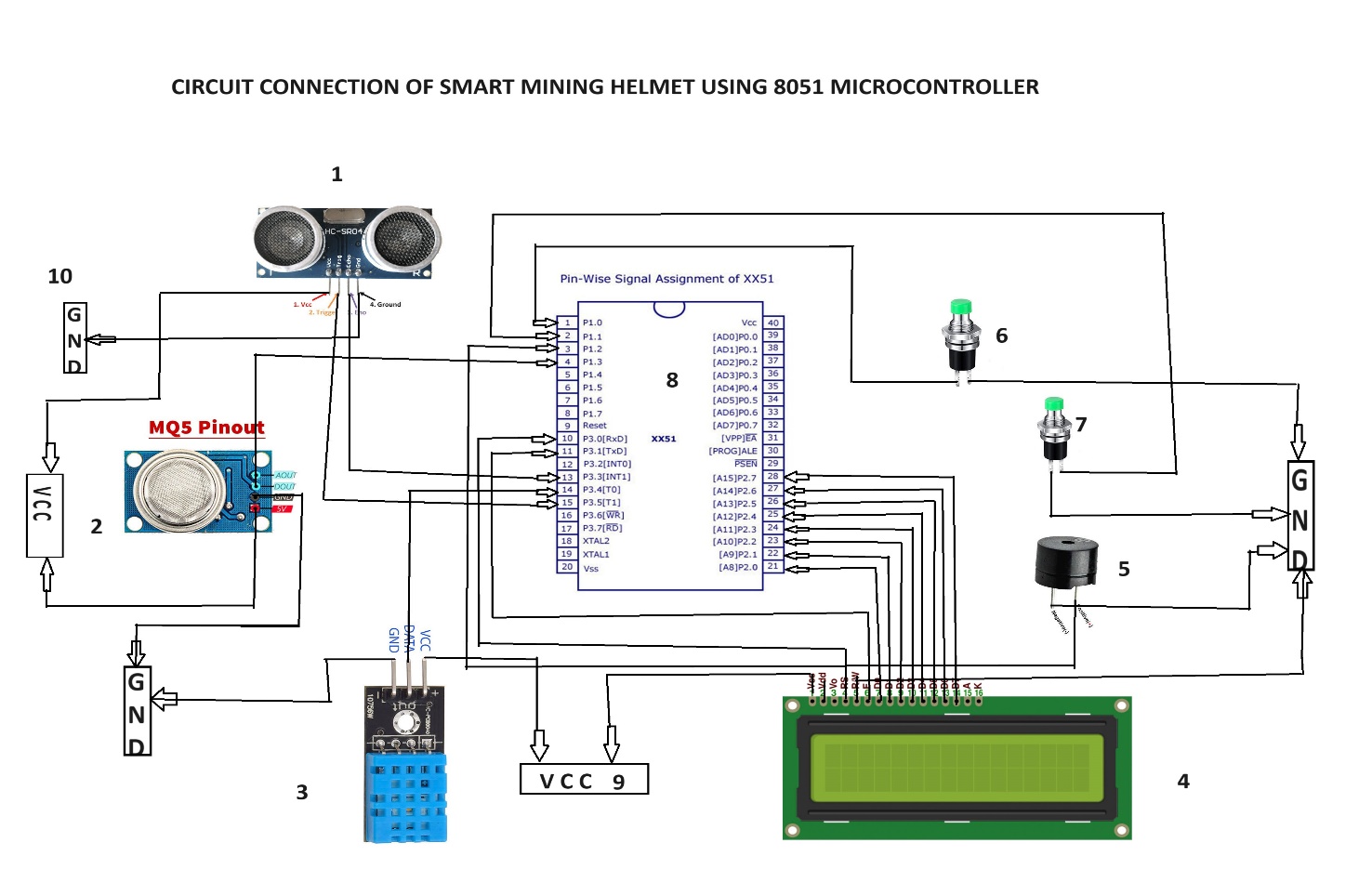
**Buzzer:**

****

* **Also known as a sounder, audio alarm or audio indicator, a buzzer is a basic audio device that generates a sound from an incoming electrical signal**

**Protective Helmet Structure:**

* **Integrates all components into a durable and miner-friendly helmet structure.**
* **Ensures comfort, durability, and adherence to safety standards.**

**CIRCUIT DIAGRAM AND PIN CONFIGURATION: **

**COMPONENTS NAME & PIN CONFIGURATION:**

1. **ULTRASONIC SENSOR - Echo pin P3^3 & Trig pin P3^5**
2. **MQ5 SENSOR - Digital pin P1^3**
3. **DHT11 SENSOR - Digital pin P3^4**
4. **LCD DISPLAY - DATA pin P2^0-P2^7, ES pin P3^0 & EN pin P3^1**
5. **BUZZER - Output pin P1^2**
6. **SWITCH1-ULTRASONIC - P1^0**
7. **SWITCH2-DHT11 - P1^1**
8. **8051 MICRO CONTROLLER**
9. **VCC**
10. **GND**

**CODEING:**

**1.MQ5 SENSOR CODE**

**#include<reg51.h>**

**sbit Led = P^3;**

**sbit sensor = P1^ 4;**

**main()**

**{**

**Led = 0 ;**

**sensor = 1;**

**while(1)**

**{**

**if(sensor == 0)**

**{**

**Led = 1;**

**}**

**else**

**{**

**Led = 0;**

**}**

**}**

**}**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**2. ULTRASONIC SENOR CODE:**

**#include<reg51.h>**

**#include<intrins.h>**

**#define lcd\_data P2**

**sfr16 DPTR=0x82;**

**sbit trig=P3^5;**

**sbit echo=P3^2;**

**unsigned int range=0;**

**sbit rs=P1^0;**

**//sbit rw=P0^1;**

**sbit en=P1^2;**

**void lcd\_init();**

**void cmd(unsigned char a);**

**void dat(unsigned char b);**

**void show(unsigned char \*s);**

**void lcd\_delay();**

**void lcd\_init()**

**{**

**cmd(0x38);**

**cmd(0x0e);**

**cmd(0x06);**

**cmd(0x0c);**

**cmd(0x80);**

**}**

**void cmd(unsigned char a)**

**{**

**lcd\_data=a;**

**rs=0;**

**// rw=0;**

**en=1;**

**lcd\_delay();**

**en=0;**

**}**

**void dat(unsigned char b)**

**{**

**lcd\_data=b;**

**rs=1;**

**//rw=0;**

**en=1;**

**lcd\_delay();**

**en=0;**

**}**

**void show(unsigned char \*s)**

**{**

**while(\*s)**

**{**

**dat(\*s++);**

**}**

**}**

**void lcd\_delay()**

**{**

**unsigned int i;**

**for(i=0;i<=1000;i++);**

**}**

**void send\_pulse(void)**

**{**

**TH0=0x00;TL0=0x00;**

**trig=1;**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**trig=0;**

**}**

**unsigned char ultrasonic()**

**{**

**unsigned char get;**

**send\_pulse();**

**while(!echo);**

**while(echo);**

**DPH=TH0;**

**DPL=TL0;**

**TH0=TL0=0xff;**

**if(DPTR<38000)**

**get=DPTR/59;**

**else**

**get=0;**

**return get;**

**}**

**void main()**

**{**

**TMOD=0x09;**

**TH0=TL0=0;**

**TR0=1;**

**lcd\_init();**

**cmd(0x01);**

**show("DISTANCE:");**

**P3|=(1<<2);**

**while(1)**

**{**

**cmd(0x8A);**

**range=ultrasonic();**

**dat((range/100)+48);**

**dat(((range/10)%10)+48);**

**dat((range%10)+48);**

**lcd\_delay();lcd\_delay();lcd\_delay();lcd\_delay();**

**}**

**}**

**----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**3. DHT11 SENOR CODE:**

**#include<reg51.h>**

**#include<intrins.h>**

**#include"lcd.h"**

**sbit DHT11=P1^1;**

**int I\_RH,D\_RH,I\_Temp,D\_Temp,checksum;**

**void delay\_40ms();**

**void delay\_40ms()**

**{**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**}**

**void Request()**

**{**

**DHT11=0;**

**delay\_40ms();**

**DHT11=1;**

**}**

**void Response()**

**{**

**while(DHT11==1);**

**while(DHT11==0);**

**while(DHT11==1);**

**}**

**int Receive\_data()**

**{**

**int i,dataD=0;**

**for(i=0;i<8;i++)**

**{**

**while(DHT11==0);**

**delay\_40ms();**

**if(DHT11==1)**

**dataD=(dataD<<1)|(0x01);**

**else**

**dataD=(dataD<<1);**

**while(DHT11==1);**

**}**

**return dataD;**

**}**

**void main()**

**{**

**unsigned int a,b,c,d;**

**initlcd();**

**cmd\_lcd(0x80);**

**Print\_String("Welcome");**

**delay\_40ms();**

**cmd\_lcd(0x01);**

**cmd\_lcd(0x80);**

**Print\_String("Humidity ");**

**cmd\_lcd(0xc0);**

**Print\_String("Temperature");**

**while(1)**

**{**

**Request();**

**Response();**

**I\_RH=Receive\_data();**

**D\_RH=Receive\_data();**

**I\_Temp=Receive\_data();**

**D\_Temp=Receive\_data();**

**checksum=Receive\_data();**

**if((I\_RH+D\_RH+I\_Temp+D\_Temp)!=checksum)**

**{**

**Print\_String("Error");**

**}**

**else**

**{**

**a=(I\_RH/10)+48;**

**b=(I\_RH%10)+48;**

**cmd\_lcd(0x89);**

**data\_lcd(a);**

**cmd\_lcd(0x8a);**

**data\_lcd(b);**

**cmd\_lcd(0x8b);**

**data\_lcd('%');**

**c=(I\_Temp/10)+48;**

**d=(I\_Temp%10)+48;**

**cmd\_lcd(0xcc);**

**data\_lcd(c);**

**cmd\_lcd(0xcd);**

**data\_lcd(d);**

**cmd\_lcd(0xce);**

**data\_lcd(0xDF);**

**cmd\_lcd(0xcf);**

**data\_lcd('C');**

**delay\_40ms();**

**}**

**}**

**}**

**------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**4.LCD DISPLAY CODE:**

**#include"header.h**

**#define lcd P2 //lcd data pin is connected to port 2 pin**

**sbit RS = P3^0; // RS pin connected to P3.0**

**sbit EN = P3^1; // EN pin connected to P3.1**

**sbit Led = P1^2; // P1^2 is connected to an LED**

**void initlcd()**

**{**

**Led = 1; // Turn on an LED at the beginning of the initialization**

**cmd\_lcd(0x38);**

**delay\_ms(5);**

**cmd\_lcd(0x0C);**

**delay\_ms(5);**

**cmd\_lcd(0x01);**

**delay\_ms(5);**

**cmd\_lcd(0x06);**

**delay\_ms(5);**

**Led = 0; // Turn off the LED at the end of the initialization**

**}**

**void cmd\_lcd(unsigned char command)**

**{**

**EN = 1;**

**RS = 0;**

**lcd = command;**

**delay\_ms(1);**

**EN = 0;**

**//delay\_ms(10);**

**}**

**void data\_lcd(unsigned char databyte)**

**{**

**EN = 1;**

**RS = 1;**

**lcd = databyte;**

**delay\_ms(1);**

**EN = 0;**

**}**

**void Print\_String(unsigned char \*message)**

**{**

**while(\*message!='\0')**

**{**

**data\_lcd( \*message);**

**message++;**

**}**

**}**

**void delay\_ms(unsigned int ms)**

**{**

**unsigned char t1;**

**unsigned int t2;**

**for(t1=0; t1<ms; t1++)**

**{**

**for(t2=0; t2<114; t2++);**

**}**

**}**

**// Function to print an integer on LCD**

**//void Print\_Integer(unsigned int num)**

**//{**

**//**

**// if (num < 10) {**

**// data\_lcd(num + '0');**

**// } else {**

**// Print\_Integer(num / 10);**

**// data\_lcd((num % 10) + '0');**

**// }}**

**//}**

**--------------------------------------------------------------------------------------------**

**INTEGRATING ALL 4 SENOSR AND DSIPLAY CODE:**

**#include"lcd.h"**

**#include"header.h"**

**#include<stdio.h>**

**sbit UltrasonicSwitch = P1^0; // Switch for Ultrasonic sensor**

**sbit DHTSwitch = P1^1; // Switch for DHT11 sensor**

**sbit echo = P3^3; // Ultrasonic Echo pin**

**sbit trig = P3^5; // Ultrasonic Trig pin**

**sbit DHT11 = P3^4; // DHT11 sensor data pin**

**sfr16 DPTR=0x82;**

**sbit sensor = P1^3; // P1^3 is connected to the output of the MQ gas sensor**

**unsigned int I\_RH, D\_RH, I\_Temp, D\_Temp, checksum;**

**unsigned char DHT11Switch;**

**unsigned int range=0;**

**void delay\_40ms() {**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_(); \_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**}**

**void Request()**

**{**

**DHT11 = 0;**

**delay\_40ms();**

**DHT11 = 1;**

**}**

**void Response()**

**{**

**while(DHT11 == 1);**

**while(DHT11 == 0);**

**while(DHT11 == 1);**

**}**

**int Receive\_data()**

**{**

**int i, dataD = 0;**

**for(i = 0; i < 8; i++) {**

**while(DHT11 == 0);**

**delay\_40ms();**

**if(DHT11 == 1)**

**dataD = (dataD << 1) | (0x01);**

**else**

**dataD = (dataD << 1);**

**while(DHT11 == 1);**

**}**

**return dataD;**

**}**

**void Display\_DHT11()**

**{**

**unsigned int a, b, c, d;**

**cmd\_lcd(0x01);**

**cmd\_lcd(0x80);**

**Print\_String("Humidity ");**

**cmd\_lcd(0xC0);**

**Print\_String("Temperature");**

**Request();**

**Response();**

**I\_RH = Receive\_data();**

**D\_RH = Receive\_data();**

**I\_Temp = Receive\_data();**

**D\_Temp = Receive\_data();**

**checksum = Receive\_data();**

**if((I\_RH + D\_RH + I\_Temp + D\_Temp) != checksum)**

**{**

**Print\_String("Error");**

**}**

**else**

**{**

**a = (I\_RH / 10) + 48;**

**b = (I\_RH % 10) + 48;**

**cmd\_lcd(0x89);**

**data\_lcd(a);**

**cmd\_lcd(0x8A);**

**data\_lcd(b);**

**cmd\_lcd(0x8B);**

**data\_lcd('%');**

**c = (I\_Temp / 10) + 48;**

**d = (I\_Temp % 10) + 48;**

**cmd\_lcd(0xCC);**

**data\_lcd(c);**

**cmd\_lcd(0xCD);**

**data\_lcd(d);**

**cmd\_lcd(0xCE);**

**data\_lcd(0xDF);**

**cmd\_lcd(0xCF);**

**data\_lcd('C');**

**}**

**}**

**void send\_pulse(void)**

**{**

**TH0=0x00;TL0=0x00;**

**trig=1;**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**trig=0;**

**}**

**unsigned char ultrasonic()**

**{**

**unsigned char get;**

**send\_pulse();**

**while(!echo);**

**while(echo);**

**DPH=TH0;**

**DPL=TL0;**

**TH0=TL0=0xff;**

**if(DPTR<38000)**

**get=DPTR/59;**

**else**

**get=0;**

**return get;**

**}**

**void Gas\_Sensor()**

**{**

**// Assuming the Gas Sensor operation**

**if(sensor == 1)**

**{**

**Led = 1;**

**}**

**else**

**{**

**Led = 0;**

**}**

**}**

**void main() {**

**//Ultrasonic\_Init();**

**initlcd(); // Call your LCD initialization function**

**cmd\_lcd(0x01);**

**cmd\_lcd(0x80); // Set cursor to the beginning of the first line**

**Print\_String("LCD init..");**

**delay\_ms(100); // Delay for 40ms**

**while (1)**

**{**

**if (UltrasonicSwitch == 0)**

**{**

**TMOD=0x09;**

**TH0=TL0=0;**

**TR0=1;**

**//initlcd();**

**cmd\_lcd(0x01);**

**Print\_String("DISTANCE:");**

**P3|=(1<<2);**

**cmd\_lcd(0x8A);**

**range=ultrasonic();**

**data\_lcd((range/100)+48);**

**data\_lcd(((range/10)%10)+48);**

**data\_lcd((range%10)+48);**

**delay\_ms(200);**

**}**

**else if (DHTSwitch == 0)**

**{**

**//cmd\_lcd(0x01); // Clear the LCD display**

**//cmd\_lcd(0x80); // Set cursor to the beginning of the first line**

**//Print\_String("Switch 2 pressed");**

**Display\_DHT11();**

**delay\_ms(100);**

**}**

**else**

**{**

**Gas\_Sensor();**

**}**

**}**

**}**

**#include <reg51.h>**

**#include <intrins.h>**

**#include <stdio.h>**

**// Function prototypes (replace these with your LCD functions)**

**void initlcd(void);**

**void cmd\_lcd(unsigned char command);**

**void data\_lcd(unsigned char databyte);**

**void Print\_String(unsigned char \*message);**

**void delay\_ms(unsigned int);**

**void data\_lcd(unsigned char);**

**void Print\_String(char \*);**

**void Print\_Integer(unsigned int);**

**void delay(unsigned int);**

**void Init\_Timer0();**

**unsigned int Measure\_Distance()**

**#include"header.h"**

**#define lcd P2 //lcd data pin is connected to port 2 pin**

**sbit RS = P3^0; // RS pin connected to P3.0**

**sbit EN = P3^1; // EN pin connected to P3.1**

**sbit Led = P1^2; // P1^2 is connected to an LED**

**void initlcd()**

**{**

**Led = 1; // Turn on an LED at the beginning of the initialization**

**cmd\_lcd(0x38);**

**delay\_ms(5);**

**cmd\_lcd(0x0C);**

**delay\_ms(5);**

**cmd\_lcd(0x01);**

**delay\_ms(5);**

**cmd\_lcd(0x06);**

**delay\_ms(5);**

**Led = 0; // Turn off the LED at the end of the initialization**

**}**

**void cmd\_lcd(unsigned char command)**

**{**

**EN = 1;**

**RS = 0;**

**lcd = command;**

**delay\_ms(1);**

**EN = 0;**

**//delay\_ms(10);**

**}**

**void data\_lcd(unsigned char databyte)**

**{**

**EN = 1;**

**RS = 1;**

**lcd = databyte;**

**delay\_ms(1);**

**EN = 0;**

**}**

**void Print\_String(unsigned char \*message)**

**{**

**while(\*message!='\0')**

**{**

**data\_lcd( \*message);**

**message++;**

**}**

**}**

**void delay\_ms(unsigned int ms)**

**{**

**unsigned char t1;**

**unsigned int t2;**

**for(t1=0; t1<ms; t1++)**

**{**

**for(t2=0; t2<114; t2++);**

**}**

**}**

**// Function to print an integer on LCD**

**//void Print\_Integer(unsigned int num)**

**//{**

**// if (num < 10) {**

**// data\_lcd(num + '0');**

**// } else {**

**// Print\_Integer(num / 10);**

**// data\_lcd((num % 10) + '0');**

**// }**

**//}**

**#include<reg51.h>**

**#include<intrins.h>**

**#define lcd\_data P2**

**sfr16 DPTR=0x82;**

**sbit trig=P3^5;**

**sbit echo=P3^2;**

**unsigned int range=0;**

**sbit rs=P0^6;**

**//sbit rw=P0^1;**

**sbit en=P0^7;**

**void lcd\_init();**

**void cmd(unsigned char a);**

**void dat(unsigned char b);**

**void show(unsigned char \*s);**

**void lcd\_delay();**

**void lcd\_init()**

**{**

**cmd(0x38);**

**cmd(0x0e);**

**cmd(0x06);**

**cmd(0x0c);**

**cmd(0x80);**

**}**

**void cmd(unsigned char a)**

**{**

**lcd\_data=a;**

**rs=0;**

**// rw=0;**

**en=1;**

**lcd\_delay();**

**en=0;**

**}**

**void dat(unsigned char b)**

**{**

**lcd\_data=b;**

**rs=1;**

**//rw=0;**

**en=1;**

**lcd\_delay();**

**en=0;**

**}**

**void show(unsigned char \*s)**

**{**

**while(\*s) {**

**dat(\*s++);**

**}**

**}**

**void lcd\_delay()**

**{**

**unsigned int i;**

**for(i=0;i<=1000;i++);**

**}**

**void send\_pulse(void)**

**{**

**TH0=0x00;TL0=0x00;**

**trig=1;**

**\_nop\_(); \_nop\_(); \_nop\_(); \_nop\_(); \_nop\_();**

**\_nop\_(); \_nop\_(); \_nop\_(); \_nop\_();**

**trig=0;**

**}**

**unsigned char ultrasonic()**

**{**

**unsigned char get;**

**send\_pulse();**

**while(!echo);**

**while(echo);**

**DPH=TH0;**

**DPL=TL0;**

**TH0=TL0=0xff;**

**if(DPTR<38000)**

**get=DPTR/59;**

**else**

**get=0;**

**return get;**

**}**

**void main()**

**{**

**TMOD=0x09;**

**TH0=TL0=0;**

**TR0=1;**

**lcd\_init();**

**cmd(0x01);**

**show("DISTANCE:");**

**P3|=(1<<2);**

**while(1)**

**{**

**cmd(0x8A);**

**range=ultrasonic();**

**dat((range/100)+48);**

**dat(((range/10)%10)+48);**

**dat((range%10)+48);**

**lcd\_delay();lcd\_delay();lcd\_delay();lcd\_delay();**

**}**

**}#include<reg51.h>**

**#include<intrins.h>**

**#define lcd\_data P2**

**sfr16 DPTR=0x82;**

**sbit trig=P3^5;**

**sbit echo=P3^2;**

**unsigned int range=0;**

**sbit rs=P1^0;**

**//sbit rw=P0^1;**

**sbit en=P1^2;**

**void lcd\_init();**

**void cmd(unsigned char a);**

**void dat(unsigned char b);**

**void show(unsigned char \*s);**

**void lcd\_delay();**

**void lcd\_init()**

**{**

**cmd(0x38);**

**cmd(0x0e);**

**cmd(0x06);**

**cmd(0x0c);**

**cmd(0x80);**

**}**

**void cmd(unsigned char a)**

**{**

**lcd\_data=a;**

**rs=0;**

**// rw=0;**

**en=1;**

**lcd\_delay();**

**en=0;**

**}**

**void dat(unsigned char b)**

**{**

**lcd\_data=b;**

**rs=1;**

**//rw=0;**

**en=1;**

**lcd\_delay();**

**en=0;**

**}**

**void show(unsigned char \*s)**

**{**

**while(\*s)**

**{**

**dat(\*s++);**

**}**

**}**

**void lcd\_delay()**

**{**

**unsigned int i;**

**for(i=0;i<=1000;i++);**

**}**

**void send\_pulse(void)**

**{**

**TH0=0x00;TL0=0x00;**

**trig=1;**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**\_nop\_();\_nop\_();\_nop\_();\_nop\_();\_nop\_();**

**trig=0;**

**}**

**unsigned char ultrasonic()**

**{**

**unsigned char get;**

**send\_pulse();**

**while(!echo);**

**while(echo);**

**DPH=TH0;**

**DPL=TL0;**

**TH0=TL0=0xff;**

**if(DPTR<38000)**

**get=DPTR/59;**

**else**

**get=0;**

**return get;**

**}**

**void main()**

**{**

**TMOD=0x09;**

**TH0=TL0=0;**

**TR0=1;**

**lcd\_init();**

**cmd(0x01);**

**show("DISTANCE:");**

**P3|=(1<<2);**

**while(1)**

**{**

**cmd(0x8A);**

**range=ultrasonic();**

**dat((range/100)+48);**

**dat(((range/10)%10)+48);**

**dat((range%10)+48);**

**lcd\_delay();lcd\_delay();lcd\_delay();lcd\_delay();**

**}**

**}**

**---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**Functionality:**

**The Smart Mining Helmet operates in three primary modes, each activated by a dedicated switch:**

**1.Ultrasonic Distance Measurement Mode (Switch 1):**

* **When Switch 1 is pressed, the microcontroller activates the ultrasonic sensor.**
* **The distance between the helmet and obstacles is measured and displayed on the LCD.**
* **This feature aids miners in navigating through confined spaces, preventing collisions.**

**2.** **Temperature and Humidity Measurement Mode (Switch 2):**

* **Pressing Switch 2 triggers the DHT11 sensor to measure temperature and humidity.**
* **The LCD displays real-time temperature and humidity data, allowing miners to a ssess environmental comfort and potential health risks.**
* **Gas Detection Mode:**

**3.** **The MQ5 gas sensor continuously monitors the air for harmful gas levels.**

**If a dangerous concentration is detected, the microcontroller activates the buzzer to alert the wearer**

**User Interface:**

**The LCD display serves as the user interface, providing clear and concise information in real-time. The data displayed includes:**

**Ultrasonic Distance: Assists in spatial awareness and obstacle avoidance.**

**Temperature and Humidity: Offers insights into the environmental conditions.**

**Gas Detection Warning: Immediate the buzzer will turn on.**

**The Smart Mining Helmet goes beyond conventional safety gear by actively monitoring and alerting miners to potential hazards. The integration of gas detection and obstacle avoidance features enhances overall safety, reducing the risk of accidents and injuries in mining environments.**

**Purpose of choosing the 8051 Microcontroller for this project is:**

**1. One of the primary reasons for selecting the 8051 microcontroller is its long-established reputation for robustness and reliability.**

**2.** **The 8051 architectures has been a stable in the embedded systems industry for decades, known for its stability and predictable performance.**

**3. The 8051 microcontroller's wide spread adoption across various industries, including automotive, aerospace, and consumer electronics, underscores its versatility and acceptance.**

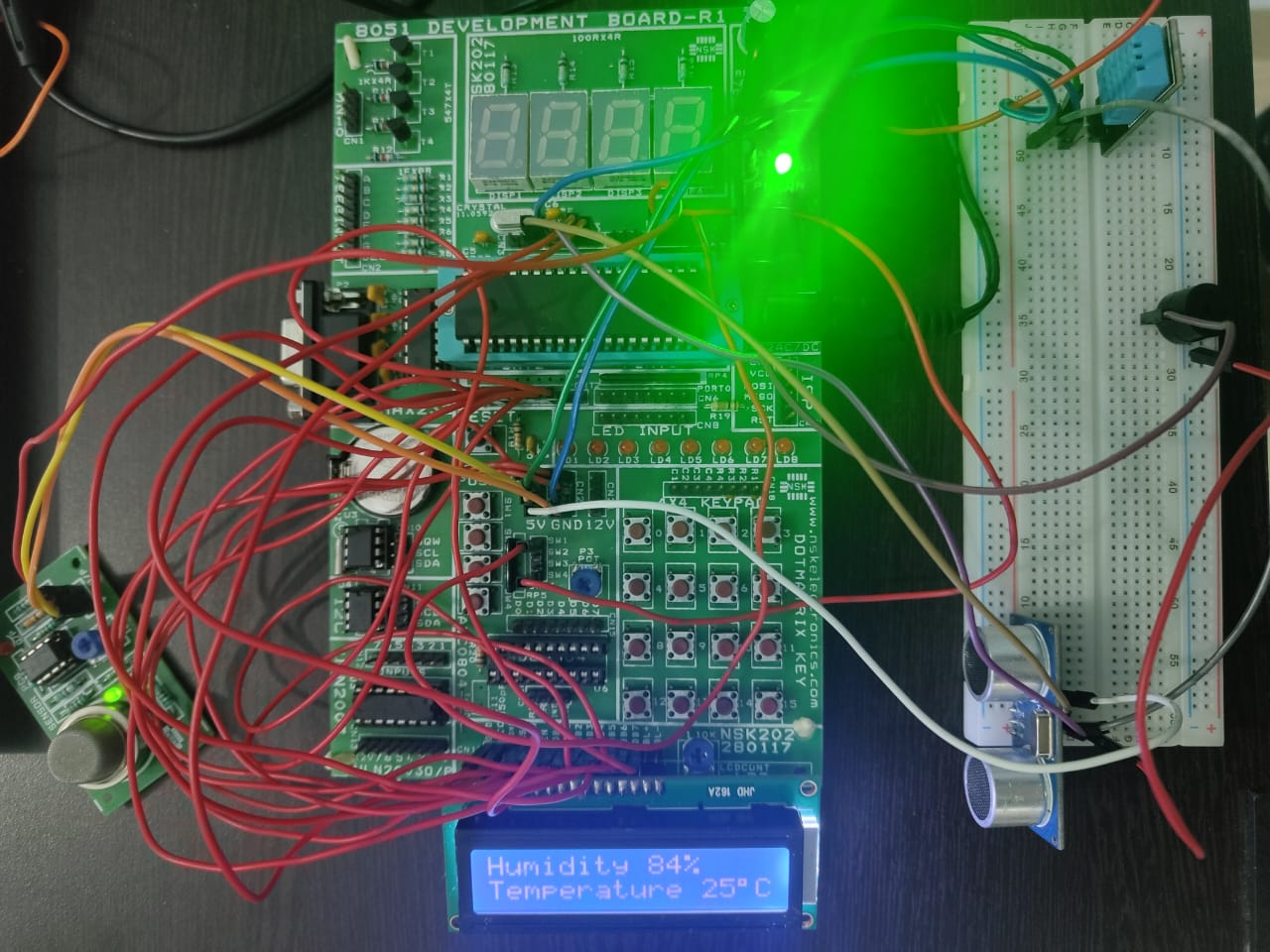
**4.** **This prevalence results in a wealth of resources, including development tools, libraries, and community support, making it an ideal choice for projects with diverse sensor integration requirements.**

**5. The 8051's ease of use allows for efficient coding, reducing development time and facilitating rapid prototyping and testing phases.**

**6. The 8051 microcontroller strikes a balance between performance and cost-effectiveness, making it an economically viable choice for the Smart Mining Helmet Project.**

**7. The availability of cost-efficient variants of the 8051 without compromising on functionality aligns well with the project's budgetary constraints.**

**RESULT/OUTPUT:**

****

**a. Output Image**

**Conclusion:**

**In conclusion, the Smart Mining Helmet with 8051 microcontroller, ultrasonic sensor, DHT11, MQ5, and LCD display is a cutting-edge solution to enhance safety and environmental monitoring in mining operations. By integrating advanced sensors and a user-friendly interface, this device empowers miners with crucial information, promoting a safer and more secure working environment. The project not only demonstrates the capabilities of modern sensor technology but also underscores its potential for revolutionizing safety standards in various industries.**