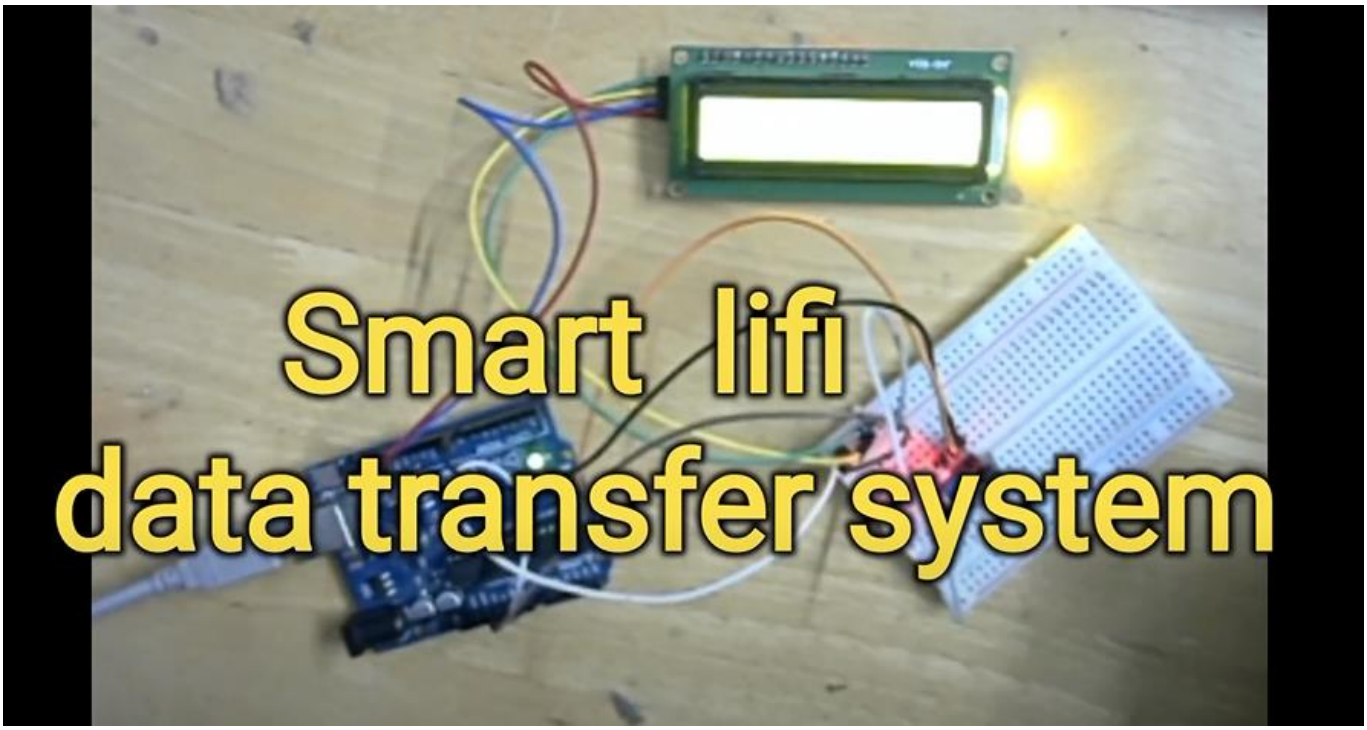
LiFi technology using data communication



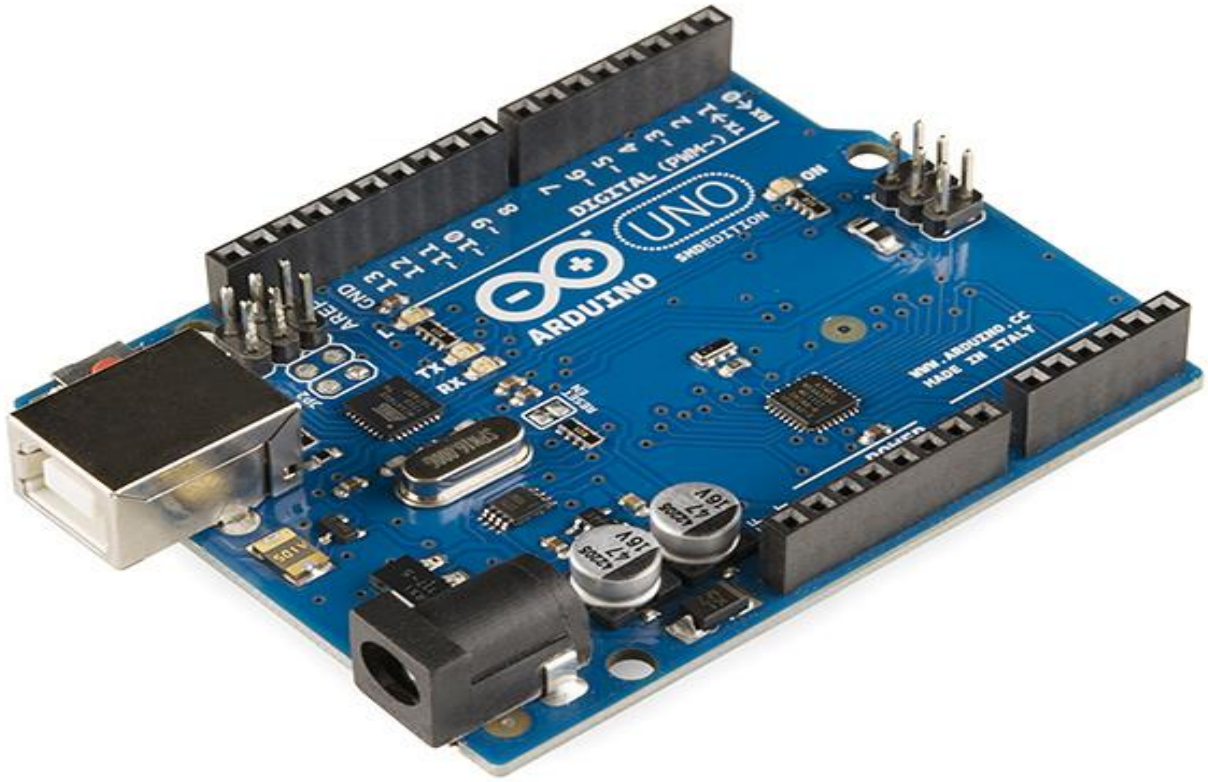
Li-Fi, which stands for Light Fidelity, is a wireless communication technology that uses light to transmit data instead of radio waves used in Wi-Fi. Li-Fi has potential applications in many fields, including accident safety systems.

Using an Arduino board, you can create a Li-Fi data transfer system for accident safety systems. Here are the basic steps to get started:

1. Set up a transmitter: Using an LED, create a transmitter circuit that can convert data into light signals. You can use any digital pin on the Arduino to control the LED.
2. Set up a receiver: Using a photodiode, create a receiver circuit that can detect the light signals and convert them back into data. Connect the photodiode to an analog pin on the Arduino.
3. Write the code: Using the Arduino IDE, write the code that controls the LED and reads the photodiode. You'll need to define the data transmission protocol and ensure that the transmitter and receiver are synchronized.
4. Test the system: Use a test program to verify that the transmitter and receiver are working correctly. You can also test the system by transmitting data between two Arduino boards.
5. Integrate with accident safety system: Once you've verified that the Li-Fi data transfer system is working, you can integrate it with your accident safety system.

Note that Li-Fi technology has some limitations, such as a limited range and the requirement for a clear line of sight between the transmitter and receiver. Therefore, you'll need to consider these factors when designing your accident safety system.

Arduino Uno



Arduino Uno is an open-source microcontroller board based on the ATmega328P microcontroller chip. It has a total of 14 digital input/output pins, 6 of which can be used as PWM (Pulse Width Modulation) outputs, and 6 analog inputs. The board also features a USB port, a power jack, an ICSP header, and a reset button.

Arduino Uno is one of the most popular microcontroller boards used in the Maker community for prototyping and DIY projects. It can be programmed using the Arduino Integrated Development Environment (IDE), which provides an easy-to-use programming environment for writing, compiling, and uploading code to the board.

With its versatility and affordability, the Arduino Uno can be used to create a wide range of projects, including robotics, home automation, and data logging. It is also compatible with a wide range of shields (add-on boards) that can extend its functionality, such as Wi-Fi, Bluetooth, and GPS modules.

Overall, the Arduino Uno is a great choice for beginners and experienced users alike who want to experiment with electronics and programming.
Ultrasonic sensors are electronic devices that use sound waves to measure distances and detect objects. They emit high-frequency sound waves, which bounce off objects in their path and are then received by the sensor. By measuring the time it takes for the sound waves to bounce back, the sensor can determine the distance to the object.

LDR MODULE

An LDR (Light Dependent Resistor) module is a sensor module that is used to detect changes in light intensity. The module contains an LDR, which is a type of resistor that changes its resistance based on the amount of light it is exposed to. The resistance of the LDR decreases as the light intensity increases, and vice versa. This change in resistance can be measured using an analog pin on a microcontroller such as Arduino.

Here are the basic steps to use an LDR module with an Arduino:

1. Connect the LDR module to the Arduino: The LDR module usually has three pins - VCC, GND, and OUT. Connect the VCC pin to a 5V pin on the Arduino, GND pin to the GND pin on the Arduino, and the OUT pin to an analog input pin on the Arduino, such as A0.
2. Write the code: Using the Arduino IDE, write the code to read the analog input pin and convert the voltage value into a light intensity value. You can use the analogRead() function to read the voltage value and map() function to convert it into a light intensity value. For example, if the LDR module has a resistance of 10K ohms in darkness and 1K ohms in bright light, then you can map the voltage value to a light intensity value between 0 and 100%.
3. Test the system: Use a test program to verify that the LDR module is working correctly. You can test the system by shining a light on the LDR and observing the light intensity value on the Arduino serial monitor.
4. Integrate with your project: Once you've verified that the LDR module is working, you can integrate it with your project. For example, you can use the LDR module to control the brightness of an LED based on the ambient light level.

Note that the sensitivity of the LDR module can be adjusted by adding a resistor in series with the LDR. The value of the resistor can be adjusted to change the sensitivity of the module to light.

I2C Interface Module for LCD(16x2)

An I2C interface module for LCD (16x2) is a module that allows you to connect a 16x2 LCD to an Arduino using the I2C protocol. The I2C protocol is a popular serial communication protocol that allows multiple devices to communicate with each other using a two-wire interface.

Here are the basic steps to use an I2C interface module for LCD (16x2) with an Arduino:

1. Connect the I2C interface module to the Arduino: The I2C interface module usually has four pins - VCC, GND, SDA, and SCL. Connect the VCC pin to a 5V pin on the Arduino, GND pin to the GND pin on the Arduino, SDA pin to the A4 pin on the Arduino, and SCL pin to the A5 pin on the Arduino.
2. Connect the LCD to the I2C interface module: Connect the 16x2 LCD to the I2C interface module using the provided header pins.
3. Install the LiquidCrystal_I2C library: In the Arduino IDE, go to Sketch > Include Library > Manage Libraries, search for "LiquidCrystal_I2C", and install the library.
4. Write the code: Using the LiquidCrystal_I2C library, write the code to initialize the LCD and display text. Here is a sample code:

Arduino code

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>


LiquidCrystal_I2C lcd(0x3F,16,2); // Set the LCD address to 0x3F for a 16 chars and 2 line display


void setup()

{

  lcd.init();                      // Initialize the LCD

  lcd.backlight();                 // Turn on the backlight
```

lcd.setCursor(0,0); // Set the cursor to the first column of the first row

lcd.print("Hello, World!"); // Print "Hello, World!" on the LCD

}

void loop()

{

}

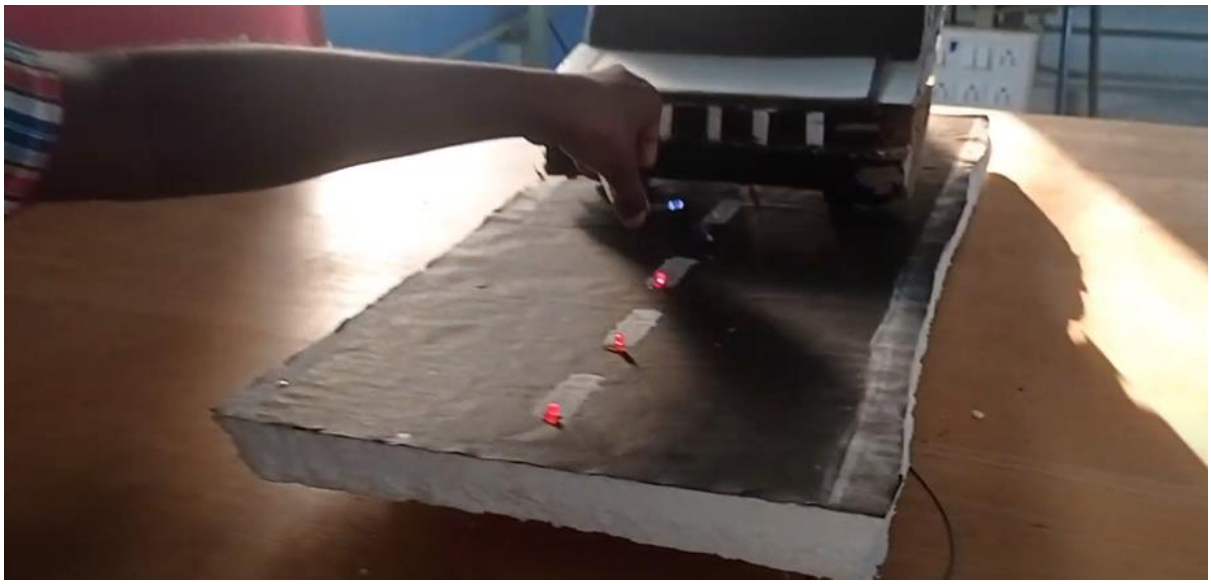
5. Upload the code and test the system: Upload the code to the Arduino and observe the text displayed on the LCD. You can modify the text to display any custom message.

Note that the I2C interface module simplifies the connection between the LCD and the Arduino by reducing the number of pins required. Additionally, the LiquidCrystal_I2C library provides an easy-to-use interface for controlling the LCD using the I2C protocol.

LCD Display

To display data on an LCD with an Arduino, you will need to follow these basic steps:

1. Connect the LCD to the Arduino: The connections between the LCD and the Arduino depend on the type of LCD you are using. Typically, an LCD has 16 pins - 8 data pins (D0-D7), 3 control pins (RS, RW, E), and 5V and GND pins. You will need to connect the data pins, control pins, and power pins to the Arduino.
2. Install the LiquidCrystal library: In the Arduino IDE, go to Sketch > Include Library > Manage Libraries, search for "LiquidCrystal", and install the library.
3. Write the code: Using the LiquidCrystal library, write the code to initialize the LCD and display text. Here is a sample code:



Arduino code

```
#include <LiquidCrystal.h>


LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Set the LCD pins to the correct pins on the Arduino


void setup() {

  lcd.begin(16, 2); // Initialize the LCD with 16 columns and 2 rows

  lcd.print("Hello, World!"); // Print "Hello, World!" on the LCD

}


void loop() {

}
```

Upload the code and test the system: Upload the code to the Arduino and observe the text displayed on the LCD. You can modify the text to display any custom message.

Note that the LiquidCrystal library provides an easy-to-use interface for controlling the LCD. You can use functions like lcd.print() and lcd.setCursor() to display text and set the cursor position on the LCD.

THANK YOU